

Twist 2



User Manual

Twist 2 for Amiga Computers

By Mermaid Group

© Copyright 1994 Mermaid Group and HiSoft.

All Rights Reserved Worldwide.

Program:

Written by Hasse Wehner, Mermaid Group.

Manual:

Written by Hasse Wehner, Karen Williams and HiSoft.

This guide and the Twist program diskettes contain proprietary information which is protected by copyright. No part of the software or the documentation may be reproduced, transcribed, stored in a retrieval system, translated into any language or transmitted in any form without express prior written consent of the publisher and copyright holder(s).

HiSoft and Mermaid Group shall not be liable for errors contained in the software or the documentation or for incidental or consequential damages in connection with the furnishing, performance or use of the software or the documentation.

HiSoft and Mermaid Group reserve the right to revise the software and/or the documentation from time to time and to make changes in the content thereof without the obligation to notify any person of such changes.

Published by HiSoft
The Old School, Greenfield, Bedford MK45 5DE UK
First English Edition, December 1994

Table of Contents

Introduction	1
Getting Started	1
System Requirements	1
Making a Backup Disk	2
Registration Card	2
The ReadMe file	3
Installing Twist	3
Running the Program	4
Install Icons	4
Installing a Printer	5
Twist and the Amiga	5
Pop-up lists	6
Object shortcuts	6
Use of keyboard in lists	7
Scrollable input boxes	7
Editor input field	8
Radio Buttons	8
Checkboxes	9
Closing edited windows	9
Twist Tutorials	11
Time Requirement	11
Tutorial 1 - Setting Up A Database	11
Creating a File	11
Creating Fields	13

Setting up Calculated Fields	15	—
Validating Fields	16	—
Entering the Data	17	—
Displaying Records	19	—
Installing an Icon	20	—
Summary	21	—
Tutorial 2 - Maintenance	21	—
Backups	21	—
Changing the Record Structure	22	—
Adding Records	22	—
Editing Records	23	—
Deleting Records	24	—
Changing the Database Structure	25	—
Deleting Fields	28	—
Summary	29	—
Tutorial 3 - Reporting	30	—
Screen Queries	30	—
Creating Formatted Reports	33	—
Editing Report Layouts	38	—
Summary	48	—
Twist Menus and Buttons	49	—
The Menus	49	—
The Project Menu	49	—
The Record Menu	59	—
The Report Menu	64	—
The Buttons or Action Gadgets	65	—

Setting up a Database	67
Database Fields	67
Editing fields	68
Field Types	68
Data type	70
Field Attributes	71
File	72
Index	73
Indexing - Overview	73
When Twist uses an Index	74
Indexing a field	75
Editing the Record Form	77
Edit form...	77
Form Design	78
Resizing	78
Moving	78
Appearance	78
Adding new objects	79
Deleting objects	79
Boxes, Backgrounds and Groups	79
Overlapping objects	79
Sorting objects	80
The Snap Gadget	82
Using the Keyboard	82
Using Objects on the Form	83
Field objects	83
Field objects	85

Text objects	88	---
Box objects	88	---
Image objects	89	---
File objects	89	---
Querying and Reporting	91	---
Screen Query	91	---
Using radio button object fields	92	---
Using check box object fields	93	---
Building a query string	93	---
Other operators	94	---
The Query buttons	95	---
About Reports	97	---
Setting up a Report	98	---
Report Destinations	98	---
New	99	---
Edit..., Delete	99	---
Selecting records for the report	99	---
Designing a Report	100	---
Adding Fields	100	---
Sorting Order	101	---
Headers and Footers	102	---
Page Layout	104	---
Adding Text	104	---
Including Calculations	105	---
Printing statistics in After sections	105	---
Linking two Files	107	---
Running the Report	108	---

Import/Export	109
Export	110
Import	110
MailMerge	111
Preparing a Text File	111
Creating a Mailmerge File	112
Running the Mailmerge	112
Saving the Mailmerge File	113
Update	115
The update window Selection	115
Add/Edit/Delete	115
File	116
Field	116
Calc	116
Save	117
Load	117
Start	117
Appendix A Expressions & Functions	119
Expressions	119
Special types	120
Function Parameters	120
The Operators	121
The Functions	125

Appendix B ARexx	135
Twist's ARexx Interface	135
Naming Ports	135
Quotation Marks	135
Appendix C Technical Support	139

Introduction

Welcome to the Twist database program, the ideal choice for anybody who wishes to organise information whether the project is a simple card-type address book or a complete, integrated sales system with relations between many database files.

This manual is written and designed to enable you to grasp the essential concepts of Twist as quickly and as easily as possible. To this end, after a short introduction, there is an extensive tutorial which should be attempted by all except the most experienced of database programmers. After the tutorial is the reference section for Twist, to be used as you would use any reference work (such as an encyclopaedia), to check on details as you need more information on how Twist works in particular situations. Experts may find that this section is all they need, but it will probably pay to be humble and at least read through the tutorial first.

Getting Started

First some general advice and some essential concepts. Please read the rest of this section carefully even if you are familiar with the Amiga and database programs; don't be too impatient, a little extra time spent taking a few precautions will be an investment for the future.

System Requirements

Twist *requires* the following equipment:

- Any Amiga computer from the A500 to the A4000.
- Workbench™ 1.3 and above.
- 1Mb of memory and two floppy disk drive.

We recommend the following if you are working with larger database files or are using pictures or sound files:

- 2Mb of memory or more. The more Fast RAM, the better.
- A hard disk or SyQuest™ removable disk mechanism. With the advent of inexpensive 270Mb SyQuest™ cartridges (approximately £55 per disk at the time of writing) we recommend this solution for its combination of flexibility, speed and value-per-megabyte.
- An accelerator for the slower Amiga computers.

Making a Backup Disk

Before installing and running Twist, it is essential to make a *backup disk* from your *master disk*.

To do this you need to format a blank floppy disk and copy the Twist master disk onto it. Do this using the Workbench Format/Copy commands, the CLI or your favourite disk copying program.

Using this backup disk means that you can keep your master disk safe from accidental damage and make a new backup should this copy become damaged; so once you have made a copy, lock up your original Twist disk away from heat, electrical interference and spilt coffee!

Registration Card

Enclosed with this manual is a registration card which you should fill in and return to us in order to register your purchase of Twist. This will entitle you to a free period of technical support and will enable us to keep you informed of future developments to our software.

For details of our technical support services, please refer to the back of this manual.

You will need to quote your serial number (found on the master disk label) to obtain technical support. You may find it useful to make a note of it here:

Serial No.

The ReadMe file

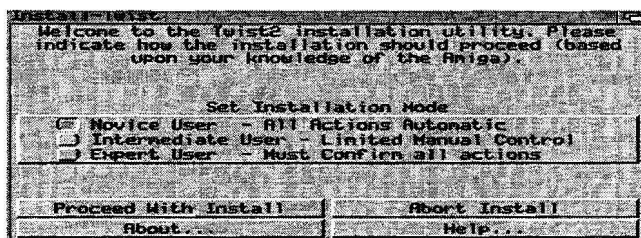
As with all HiSoft products, Twist is being improved and updated continually and the latest details may be found in the ReadMe file on the disk. It is important that you read this to find out about any new features or corrections to the manual.

You can look at the ReadMe file by double-clicking on its icon from the Workbench™ and following the instructions.

Installing Twist

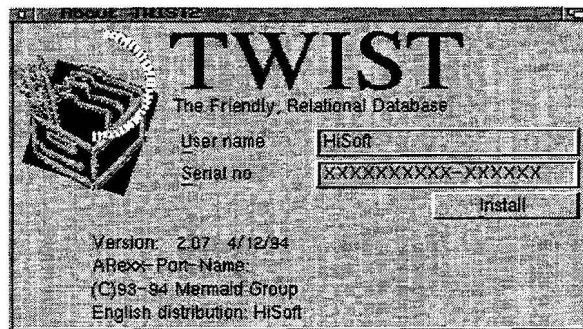
Note: The Twist installation disk is not bootable, floppy users must boot from a Workbench disk before running the Twist installation program. The floppy installation requires a copy of your original Workbench disk and a copy of the Twist Install disk, both disks will be altered during the installation.

Run the Twist installation program by double clicking on the Install-Twist icon from Workbench and follow the on-screen prompts.



the Install screen

At the end of the installation the registration program will be run, the Twist disk must *not* be write protected (floppy only) as the registration program will need to write your name and serial number into the Twist program.

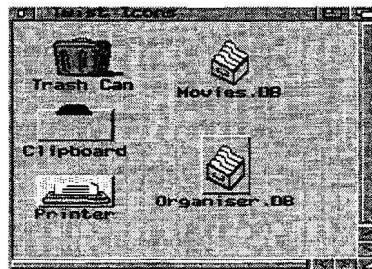


the registration requester

Running the Program

Before running Twist you will need to reboot you machine. Floppy users should place the Workbench disk altered by the Twist installation in DF0: and the registered Twist disk in DF1:, DF2: or DF3:.

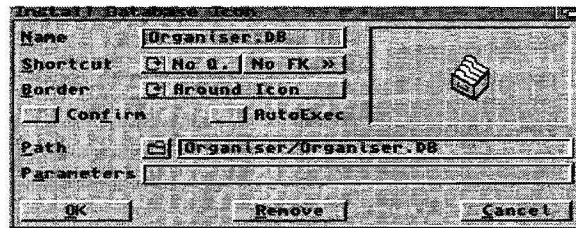
To run Twist simply double click on the Twist2 icon or type TWIST:TWIST2 from the CLI or SHELL.



the Twist Icons window

Install Icons

In addition to the Clipboard, Printer and Trashcan icons within this window, you can install appropriate icons for any files you create, by selecting the **Project** menu at the top of the screen (press the right mouse button to see the menu) and then releasing the mouse button on the **Install icon...** option. Using the file requester that appears, locate or type in a file name and select **OK**; the *Install Database Icon* requester will show up to allow you to customise this icon.

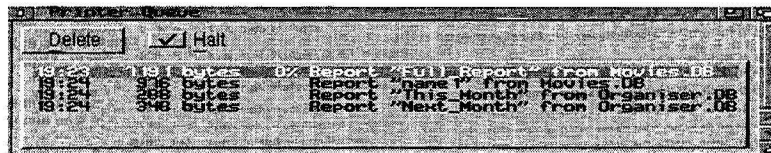


the Install Database icon requester

This requester allows you to install or remove an icon, assign a keyboard shortcut to it, give it a border, a name, a path, various parameters and also to choose whether confirmation should be given before opening the file. This will be discussed in more detail later but it serves to show the flexibility and the degree of customisation available within Twist.

Installing a Printer

Printer support is set by the printer preferences program from Workbench. Double-clicking on the Printer icon on the Twist Icons window will display the printer queue requester.



the Printer Queue requester

Twist and the Amiga

This section explains how Twist uses the Amiga user interface. Every effort has been put into making Twist as simple and easy to use as possible and it operates very much as a standard Intuition application. However, some users may not be familiar with the latest improvements to the operating system which are available on newer Amiga computers; in addition, Twist has some new ideas of its own...

Pop-up lists

To select a field name or file, Twist often uses a *pop-up* display. Such a list may contain many items only a few of which will be visible at any given time. If more items exist, then the first and last items in the list are supplemented by a small arrow at the top or bottom of the list as appropriate.



Move the mouse pointer above or below the pop-up to scroll the list. To an extent, the speed of the scroll is controlled by the distance of the pointer from the edge of the list.

Usually a pop-up is opened by clicking a mouse button in a requester or window. If you click the button with a single short click, i.e. click and release the mouse button rapidly, the pop-up list remains open until the you select and click again. If you do not wish to select an item, the pop-up can be abandoned by clicking outside the pop-up frame.

If the pop-up was opened by clicking the button without releasing the mouse, a pop-up item is selected by moving the mouse to the desired item and then releasing the mouse button. This method might be quicker for the experienced user.

A pop-up may also be used from the keyboard; for example. the cursor keys Up, Down, Shift-Up and Shift-Down move the block cursor up, down, to page up the list and to page down the list respectively. Return closes the pop-up. Esc abandons the pop-up and will ignore any selected item. The full range of shortcuts are given below.

Object shortcuts

Most objects in windows and requesters can also be activated from the keyboard. Such items have a letter underlined; hold down either Amiga key and press the key for the letter that is underlined.



In the example above, it is only necessary to hold down the right-Amiga key and then press the S key to activate the function. Of course, this may also be effected by clicking the mouse on the box itself, but experienced user may find the mouse much slower to use.

Use of keyboard in lists

Whenever Twist displays a list, a set of keystrokes can be used to move the cursor through the list. The list is automatically scrolled if the cursor is moved outside of the displayed section.

Cursor Up	Moves the cursor one item up.
Cursor Down	Moves the cursor one item down.
Shift-Up	Moves one page up.
Shift-Down	Moves one page down.
Shift + right button	Moves to the first item.
Shift + left button	Moves to the last item.

The above keys are always supported; however, depending on the context of the list, other keys may also be used:

Return/Enter	To edit the item at the block cursor; this is equivalent to double-clicking.
Delete	To delete the item at the block cursor.
	To create a new item.
Space bar	To mark the item (only in the database window in list mode).
Shift space bar	To invert the marked status of an item.

Scrollable input boxes

When entering strings that are allowed to be larger than the width of the input box, Twist automatically scrolls the text within the field so that the input cursor always remains visible. The following keys can be used to move rapidly from one end of the text to the other.

Shift-Cursor Left	Move to the start of the string.
Shift-Cursor Right	Move to the end.

These will be used mainly when entering query strings and calculation formulae which may be up to 250 characters in length and when entering long text fields in records.

Scrollable input boxes can be distinguished from fixed length boxes by the fact that they are not filled with underscores to indicate their size.

Editor input field

Keyboard commands for editor input field in record form.

Cursor keys	Moves cursor 1 position
Shift + cursor left/right	Moves the cursor to the beginning/end of line
Shift + cursor up/down	Moves the cursor a page upwards/downwards
Delete	Deletes the character at the cursor position
Backspace	Deletes the character to the left of the cursor
Ctrl + cursor left/right	Moves the cursor 1 word
Ctrl + cursor up/down	Move to the beginning/end of the text
Ctrl + delete	Deletes the current line.
Ctrl + X/C/V	Cut, copy or paste a marked block.
Ctrl + T	Marks the entire line. If a block is already marked the line underneath the marked line is added to the selected line.
Ctrl + N	Marks the whole text.
Ctrl + 1	If a block is marked then the text will be made bold or normal depending upon its current style. If no text is marked then all text entered after the cursor will be bold.
Ctrl + 2	As above but for Italic
Ctrl + 3	As above but for Underline

Text can also be marked using the mouse, press the left mouse button and move the mouse over the section of text you want to mark. To add to that area hold the shift button down while dragging over an unmarked area. To only mark a single word double click on it with the mouse.

Radio Buttons

Radio buttons provide a convenient way of entering information into a field from a pre-defined list of choices. Apart from allowing fast data entry, they also prevent the user from entering invalid or mis-typed information since the database will automatically insert the numerical value or text string into the required field.

☐ None
☐ Family
☐ Friend
☐ Contact
☐ Other

The selected item from the list is denoted by the black circle. The user may select a item by clicking on the required circle or its text label.

Checkboxes

Checkboxes are used as on/off switches or as yes/no selections on a database entry form. Unlike the radio buttons where a field may have many options, only one checkbox can be attached to each field.

Vat Registered ☐
Send Invoice ☒
Mailing List ☒

check boxes

Each option has a square to the left of the field's name. If the item is on or selected, then a cross is shown inside the square to denote its status. The data that is entered into the relevant field by Twist is either a numerical 0 or 1 to denote a False (unselected) or True status, respectively.

Closing edited windows

Some windows, like those used for editing the field or form layouts, can display the letters * before the within the window title. This shows that the contents of the window have been altered since it was opened:

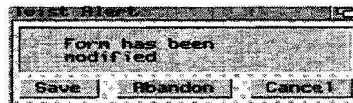


Before Edit



After Edit

Before the window is closed, Twist will ask if you wish to save the changes which have been made. This will provide you with the option of accepting the changes or of returning the window to its original condition.



For database windows however, this is not the case. When editing records, each is immediately saved to the database file to prevent an accident. If you close the window while editing a record, those changes will be lost.

Twist Tutorials

This tutorial is divided into four parts. Part 1 explains how to set up a database and enter the initial data. Part 2 is about the day-to-day running of the database. Part 3 is about running reports. Putting it all together will explain all the basic steps involved in using a database program.

In order to proceed we will assume that we work in a small business which supplies unusual and exotic items to vets and zoos.

Time Requirement

Part 1 of the tutorial will take approximately one hour to complete. Part 2 will also take an hour and Part 3 will take about an hour-and-a-half. In addition, Part 3 ideally requires the use of a printer.

Tutorial 1 - Setting Up A Database

This section of the tutorial will show you how to create a database and how to enter data. You will also learn how to install an icon representing this database file onto the Twist Icons window.

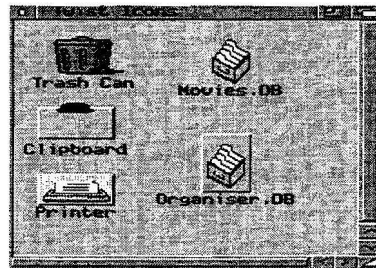
To start with you need to have installed Twist onto either a hard disk, or a floppy disk. If you have not done so, see *Getting Started* at the beginning of the manual.

Creating a File

Once you have your working copy of Twist installed, you are ready to start creating your first database. The database we are going to set up is designed to record details of orders placed by our customers, so that we can keep track of what goods we have supplied in each trading period. We need to decide which fields we have to include.

Obviously we need a description of the goods and the quantity sold, also some way of identifying to which customer we are selling them. Then we need to know the unit price and whether or not it is subject to VAT. From these items of data, we can calculate the total value of the order.

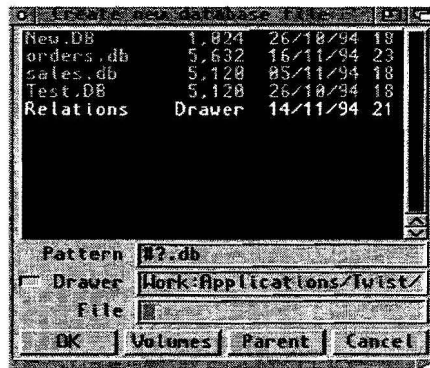
Now you can start. Open whichever disk drive contains Twist and double-click the mouse on the Twist program icon. This will give you the Twist program screen as pictured here:



Starting Twist

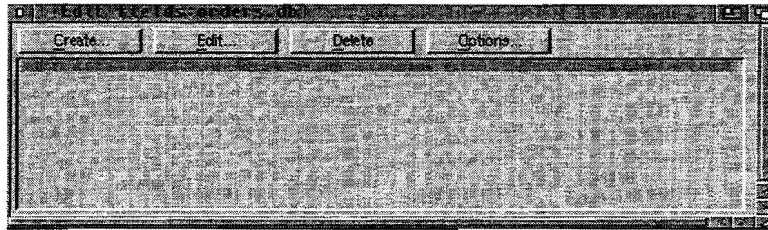
Hold the right mouse button down and move the mouse pointer to the top left of the screen; you should see the Project menu drop down. Select New from the menu by moving the mouse until New becomes highlighted and letting go the button.

You will then see a file selector:



the New Database file selector

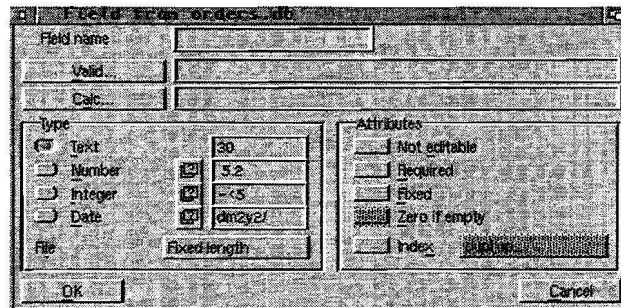
Type in `orders.DB`. Click on `OK`, or press `Return`. `Twist` will then respond by opening a new database file and displaying the *Edit fields* form thus:



the Edit fields window

Creating Fields

The blue line is the *line cursor* and shows where the next field will be created. To create a field, click on `Create` or press `Amiga-C` or `Return`. You will now see the field definition requester.



creating a field

The first thing to do when creating a field, is to give it a name. The field name can be up to 15 characters long and can contain letters and numbers but not spaces or hyphens (the first character must be a letter). If the name ought to be two words, it is usual practice to use the underline '_', to separate the words.

For the field name type in `Quantity`. Now to decide the field *type*: in some companies which normally deal in cases, but may occasionally sell part of a case, the field type would be number, to allow for decimal places. Because this company deals in single items, there can't be part of an item sold, so the field type should be an integer. Select `Integer` from the screen by left-clicking in the box to the left of the word `Integer`.

To the right of the integer type is the number 5. This refers to the length of the field, a field 5 digits long will hold any value up to 99999, which is fine for our needs.

As it makes no sense to record an order without the quantity, this field can be given the attribute **Required**; so select **Required** by left-clicking in the box. Now click on **OK** to save the field and get ready for a new field.

We now want to create a field which will hold a description of the goods sold. This time, type in **Description**, then select **Text**, to make it a text field. Do this by holding the left-Amiga key down and hitting the **T** key - this is an alternative to clicking in the box.

Now hit the **Tab** key until the cursor is in the box to the right of the word **Text** (this box should currently contain the number 30); using the **Tab** key is an alternative to clicking in the box. Now move the cursor to the right of 30 (use the right-cursor key or click); delete 30 using the **Backspace** key and enter 50. Click on **OK** to save that field.

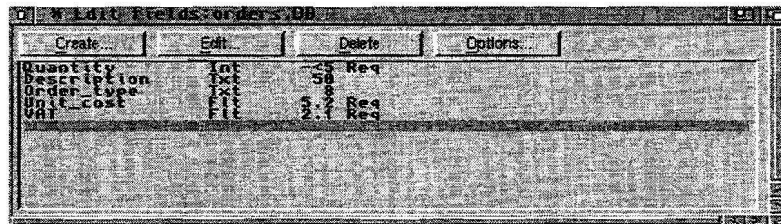
We want to record whether most orders come in by phone or letter, so we will create a field called **Order_type**. Enter this name and make it a text field 8 characters long.

Now set up the following fields:

Unit_cost	Number	5 digits	2 decimal places	Required
VAT	Number	2 digits	1 decimal place	Required

Note that, when specifying a number, 5.2 means 5 digits and 2 decimal places, 8.0 means 8 digits, no decimal places etc. So enter 5.2 and 2.1 respectively for the above fields.

Now click in the close box at the top left of the window to exit the *field definition* form, revealing the list of fields so far created in this database.



the field list

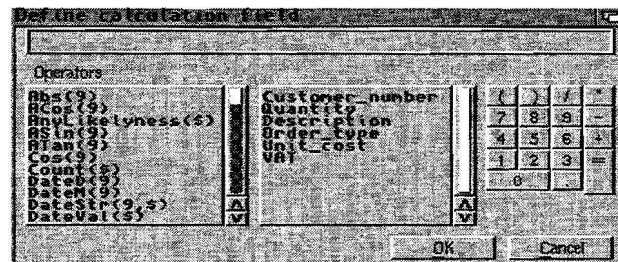
Move the line cursor back to the top of the list with the mouse or the cursor-up key, so that it is on **Quantity** and create the field **Customer_number**, a required Integer field 5 digits long, select OK and exit the definition window as before.

This field will be created where the cursor is and so will then become the first field in the list. You don't necessarily have to create the fields in the order you want them to appear on the screen.

Setting up Calculated Fields

We now have six basic fields in our database, **Customer_number**, **Quantity**, **Description**, **Order_type**, **Unit_cost** and **VAT**. We are going to add two more fields, which will be calculated from the data in the fields we have already created. These will give the value of the order without VAT and the value of the order with VAT. We will call the first one **Value** and place it between **Unit_cost** and **VAT**.

Move the block cursor to **VAT** and select **Create**. Type in **Value** for the field name. Select **Calc**. This will bring up the *Define calculation field*, requester which looks like this:



the Define Calculation field requester

Note that there is the **Operators** list, which contains more entries than those shown in the window and can be scrolled through, using the up and down arrows on the slider at the right-hand side of the box.

There is also a field list, which should contain the names of the six fields you have created so far. And there is a numeric keypad. All of these items can be selected using the mouse, to build the required formula for the calculation, or the formula can be typed in from the keyboard.

Select **Quantity** from the field list, the multiplication sign (*) from the numeric keypad and **Unit_cost** from the field list. Click on **OK** or press **Amiga-0**. Make the field a number field with 8 digits and two decimal places (8.2). Also give it the attribute **Not Editable**, as there is usually no point in letting the program do your calculation only to have someone change it accidentally later on (this attribute is checked by default for calculated fields). Confirm with **OK** and exit the definition form.

The second calculated field will be called **Total** and needs to be placed after **VAT**. Assuming that the **VAT** field will contain either 0 or 17.5, for the appropriate rate of VAT charged on each item, the formula for this field is:

$$(\text{Value} * (\text{VAT} / 100)) + \text{Value}$$

That is, the value of the order multiplied by the VAT figure divided by 100 to give the percentage of the value that has to be added, plus the original value to give the total inclusive of VAT.

Move the line cursor to the end of the field list, select **Create** and enter the appropriate field name and field names/numbers/symbols from the keypad to enter this formula. Like **Value**, **Total** should be a number field of 8 digits and two decimal places and should be **Not Editable**.

Validating Fields

We have made **Quantity** a required field in order to be certain that data is entered. It would also make sense to be sure that when a quantity is entered, it is at least 1. To do this we have to *validate* the field.

Move the block cursor back up to **Quantity** and click on **Edit**. Then select **Valid**. This gives you the *Define calculation field* requester. The formula we want in this case is **Quantity** is greater than or equal to 1. This can be typed straight into the formula box, or you may select **Quantity** from the field list, **>=** from near the bottom of the **Operators** list (use the slider at the side of the box to move to the bottom of the list) and **1** from the numeric keypad. Click **OK** and, when back on the field form, click **OK** again.

If you had tried to validate the field when you first created it, without saving it first, the field list would have been empty and even if you had typed in the word **Quantity**, the program would not have recognised it, because the field name was not in the program memory.

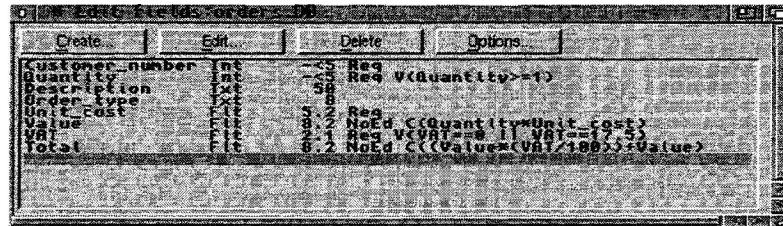
We also need to validate the field **VAT**, to ensure that only 0 or 17.5 can be entered. This validation can be changed if the rates change, but in the meantime, can save many potentially expensive typing errors!

Double-click on the VAT field (a shortcut for Edit). Select Valid and enter the formula:

VAT==0 || VAT==17.5

The symbols for *equals* and *or* are near the bottom of the Operators list and look like this; == and ||. Click OK and then OK again.

Your finished form should look something like this:

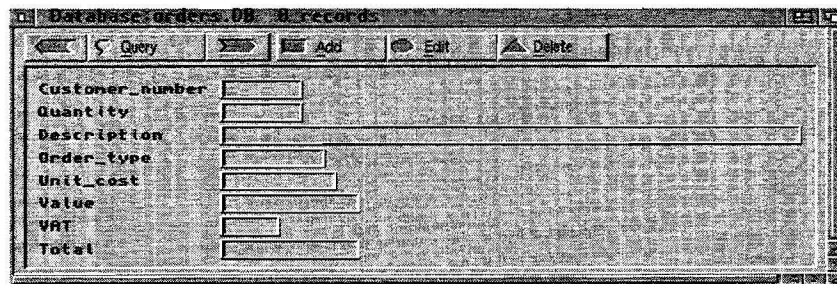


the final Field List

Entering the Data

You should now have eight fields, Customer_number, Quantity, Description, Order_type, Unit_cost, Value, VAT and Total.

So you are ready to start entering some data. Click on the close box in the top left-hand corner of the screen and this will take you to the record form. There is a blank record on the screen, with the field names in a column down the left-hand side and empty boxes the size of each field next to them. Re-size this window by left-clicking on the triangle at the bottom right of the window, holding the button down and dragging until the window looks like this:



a blank record

To enter data you are going to be adding a new record. So you need to select Add. This gives you the data entry form, which looks like this:

The screenshot shows a window titled "Database - Dec 98 - Customer number". At the top are three buttons: "Save", "Cancel", and "Duplicate". Below these are several input fields with labels to their left: "Customer_number", "Quantity", "Description", "Order_type", "Unit_cost", "Value", "VAT", and "Total". Each label is followed by a text input box. The "Description" box is significantly larger than the others. The "Value" and "Total" boxes are wider than the "Unit_cost" box.

data entry

The cursor should be positioned in the first field. Type 100 and press Return, Tab or the cursor down key to move to the next field. Enter 4. In Description, put Llama carts. Enter Phone in the Order_type. The Unit_cost is 115.25. When you press Tab this time, the cursor will skip a field and go to VAT. At the same time, Value and Total will be filled in with the product of 4 (Quantity) multiplied by 115.25 (Unit_cost). Enter 17.5 in VAT and press Return. The Total field is now completed with the addition of VAT.

Now save the record, by selecting Save. The record disappears from the screen and the blank template remains, ready for the next record to be entered.

Now enter the following records, remembering to select Save at the end of each one.

Customer_number	200
Quantity	24
Description	Pair Ram's Horn Curlers
Order_type	Letter
Unit_cost	2.50
VAT	0

Customer_number	100
Quantity	20
Description	Monkey Tyre Swings
Order_type	Phone
Unit_cost	15.65
VAT	17.5

Customer_number	200
Quantity	6
Description	Bull Nose-Rings

Order_type	Phone
Unit_cost	2.25
VAT	17.5
Customer_number	300
Quantity	4
Description	Dog's Hair Curling Tongs
Order_type	Letter
Unit_cost	24.00
VAT	17.5
Customer_number	400
Quantity	50
Description	Small Animal Toenail Clippers
Order_type	Phone
Unit_cost	4.20
VAT	0
Customer_number	100
Quantity	6
Description	Giraffe Neck Mufflers
Order_type	Letter
Unit_cost	18.60
VAT	0

You should now have seven records in your database. There are two ways to leave this screen. Firstly you can click on the top left-hand corner, but this takes you out of this database file and back to the Twist program screen. In order to stay in the database, but leave the data input form, you need to select **Cancel**.

Displaying Records

You will now see the record form again, displaying the last record you entered. This is called *Record mode* and means that you view one record at a time on the screen.

To browse through the database, left-click on the arrow icons either side of the **Query** button. Alternatively use the right/left cursor keys.

There is another mode available in the record screen and that is *List mode*. To display the records in a list, select the **Record** menu (hold the right mouse button down and move the pointer to **Record**), then choose **List** mode. Do that now and see how the display changes.

Custom	Quantity	Description
100	4	Llama carts
200	4	Fair Boes's Horn Curlers
100	20	Monkey Tote Swings
200	6	Bull Nose Rings
300	4	Dog's Hair Curling Tongs
400	20	Small Animal Tooth Clippers
100	6	Giraffe Neck Mufflers

Table mode

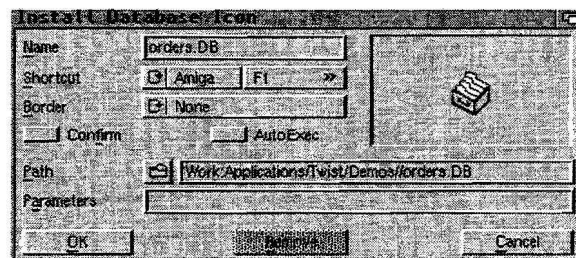
Instead of seeing the whole of one record, you can now see the first three fields of all the records. To see the other fields, move the horizontal slider at the bottom of the form.

In a larger database, not all the records would be on screen at the same time and you could scroll up and down the list with the vertical slider at the right-hand side of the form. To move the outline line cursor from one record to the next you can select the record with the mouse, or use the arrow icons next to Query. You can also use either the up/down or the left/right cursor control keys.

Installing an Icon

Now that you have your database, you want to be able to access it easily from the Twist program screen, so you need to install an icon. Click on the top left hand corner of the record form. This saves the database and returns you to the Twist program screen.

Select **Install icon** from the **File** menu. You will then see the file selector and **orders.DB** should already be in the selection; if not, select it. Press **Return**, or click on **OK**. You will then see the *Install Database Icon* requester and the current icon should be a card file icon.



installing an icon

If not then, using the slider at the side of the icon box, find the icon that says **Database**. If you wish to browse through the icons available, then do so. There are eleven of them. When you are ready, click on OK.

The Twist program screen will then be displayed, with an icon that says **orders.DB**. Now you can open the **orders.DB** database straight from this screen, by double-clicking on this icon.

Summary

In this section of the tutorial you have learnt:

- How to create a database.
- How to use calculated and validated fields.
- How to enter data.
- Two different display modes for database records.
- How to install an icon.

Tutorial 2 - Maintaining the Database

This section of the Tutorial will build on the work done in Part One and will cover the functions necessary for daily use of the database.

We will be looking at adding more records, changing data already entered and deleting obsolete records. We will also be looking at changing the structure of the database by adding, editing and deleting fields.

We will start, though, by looking at a part of regular database maintenance that is often undervalued, that is taking backups.

Backups

What are backups and why do we bother with them? Quite simply, a backup is a spare copy of any file, kept safely somewhere in case anything happens to the original.

Obviously, to be of any value, a backup needs to be kept up to date and so the frequency with which backups are taken will vary from task to task, depending on how often changes are made to the data. Daily backups are common where a large number of additions, changes etc. are carried out each day. At the very least weekly backups will be necessary.

You should have more than one set of disks used for taking backups. They need to be clearly labelled and the different sets used in rotation. For daily backups use one set of disks for each day of the week, for weekly backups, use four sets.

Once the backups have been taken they need to be checked, to see that they actually contain valid data and then be stored somewhere away from your computer system, in case of fire, flood or similar disasters.

In order to acquire the habit of good database management, take a few moments now to take a backup copy of the `orders.DB` database you created in part one of the tutorial (you should, of course, already have the original Twist program disk safely stored and be using a copy).

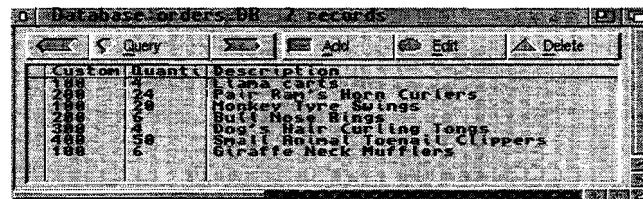
Changing the Record Structure

Once a database has been created it will almost certainly be necessary to add records or edit existing ones at some stage. This is how we go about making basic changes to our data.

Adding Records

During the life-time of any database, no matter what its function, new data will have to be added. In this instance, as new orders come in from customers, they will be added to the database.

To add a new record, open Twist and double-click on the `orders.DB` icon on the Twist Icons window. This will give you the record form as shown here:



Custom	Quantity	Description
100	4	Llama Carls
200	24	Fast Rap's Rock Curlers
100	20	Monkey Tyre Springs
200	6	Bull Nose Rings
200	4	Dog's Hair Curling Irons
200	50	Swiss Reinal Teenie Clippers
100	6	Giraffe Neck Mufflers

the orders database in List mode

Select Add. This gives you the data entry form exactly as before. Now add the following record:

Customer_number	300
Quantity	1
Description	Parakeet Travelling Vanity Set
Order_type	Letter
Unit_cost	53.45
VAT	17.5

In just the same way as when you were entering the data initially, select **Save** to save the record, then **Cancel** to return to the record screen . You should now have eight records.

There is no difference between adding records immediately after creating the database and adding them at a later date.

Editing Records

We are now going to look at making changes to existing records.

Just suppose that customer number 100 rings up and says that they have ordered 4 llama carts, but now one of their llamas has died, so they only want three! First you need to find the record for the llama carts. With only eight to choose from, it is easy to find by inspection; in a database with hundreds of records you would use a screen query to find the record. We will be covering the use of screen queries in the next part of the Tutorial.

If not already in list mode (which you should be), change the display to list mode, (select the **Record** menu and then **List mode**, or type right-Amiga-L) and move the cursor to that record. It happens to be the first one in the list.

Once that record is highlighted by the cursor, select **Edit**. This gives you the data entry form with the record to be edited in place. Use the up and down cursor keys, the **Tab** key or the mouse to move the cursor to the field you want to edit, in this case **Quantity**. Delete the 4 and enter a 3 instead. Press **Return** and see the values in the calculated fields change. Select **Save** to save the changes you have made. When editing, **Save** returns you to the record form , unlike adding records, when you remain in data entry.

Obviously you can make more than one change at a time to a record. Let's suppose that customer number 300, who ordered the curling tongs, rings up to cancel the order. You've already got them in stock and rather than have them sitting there, gathering dust, you decide to try to sell them elsewhere. Finally, customer number 400 agrees to take them on trial, but only if you reduce the price by ten per cent!

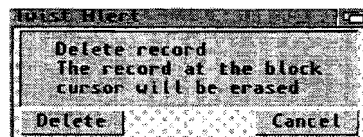
First, move the cursor down to the record for customer number 300. (the fifth one down). Select **Edit**. Change the customer number to 400 and the unit cost to 21.60. Press **Return** to check that changes take place in the calculated fields. When you are happy that the record is correct, save it.

The only fields that can't be edited are those that have been given the attribute **Not Editable** when they were created, in this case **Value** and **Total**. However, if a great deal of editing is required, it may be easier to delete the record and add a new one, rather than deleting the contents of each field, then entering new data.

Deleting Records

Just as we sometimes need to make changes to records, so we might also need to remove them from the database. The **Delete** function is as straightforward to use as editing. Suppose that customer number 300 has gone out of business. Obviously, we are no longer going to supply them with any goods, so we need to remove their order from the database.

They only have one item outstanding and that is the Parakeet Travelling Vanity Set, the last record we added. Move the cursor onto that record now and then select **Delete**. The message appears:



deleting a record

This warning gives you the chance to change your mind so, if you suddenly realise that the cursor is not in fact on the record you are intending to delete, select **Cancel**. Otherwise go ahead and remove the record by selecting **Delete**. The record has gone and we are back to seven records again.

Changing the Database Structure

The topics we have covered already, adding new records, editing existing records, deleting old records, are the main daily functions carried out on any database.

It is sometimes necessary to change the structure of the database itself, by adding, editing or deleting fields. It is not advisable to carry out these activities unless absolutely necessary and careful thought needs to be given to the original design of the database in order to avoid the need for later changes as far as possible.

In the example we are working with, the database has been designed to need changes in order to illustrate how these are carried out and also hopefully point out the type of alteration that could have been avoided with a little more forethought!

Adding New Fields

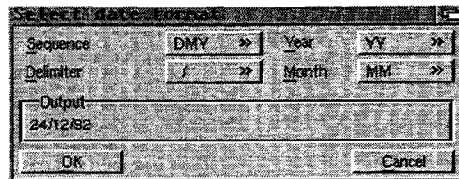
You have set up your orders database and are quite pleased with the way it is progressing. New data is being added daily and it looks as though it is going to be a real help at the end of the quarter, in evaluating sales figures and so on. Then the boss comes in, annoyed because he has had yet another complaint about slow delivery! He wants to know what you can do to help pinpoint the problem, can't your wonderful computer system help?

Well can it? If you'd known about this problem to start with, then you could have added it into the design. Is it too late? No, we can still add some more fields.

Select the List Mode and then choose **Edit fields** from the Project menu. This gives you the *Edit fields* requester which will be familiar from when you first created the database.

To create a new field, first move the cursor to the bottom of the field list. Then select **Create**.

The first new field you are going to create will be a date field to store the date that the customer placed their order. We will call this **Order_date**, so type that in for the field name. Then select **Date**. Now, left-click on the *question* icon to the right of **Date** to bring up the *date format* requester.



defining a date field

This way you can choose how you want your dates displayed. The default is DD/MM/YY (DMY), but the day, month, year can be in any sequence, the year can be either two or four digits and the month can be either two digits, the full name, or the name shortened to three letters. The delimiter between the day part, month part and year part of the date can be one of /, - or .. When you have chosen the format you want, click on OK.

It would be useful to make this field a required field, so that the date has to be entered for each record. Unfortunately it is not advisable to do this at this stage, because there are records already existing which do not have any data in this new field. If it is possible to go through and complete this field in each record in the database, then it will be possible to later edit it and make it a Required field.

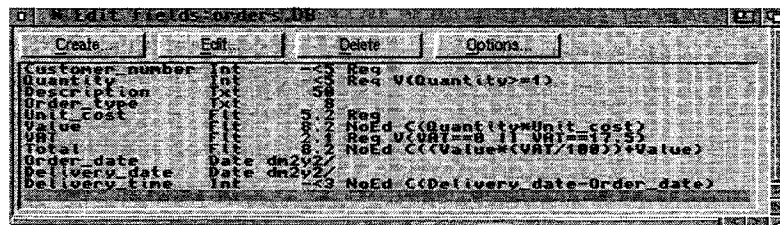
Click on OK to save this field and create another one. This will also be a date field and we will call this one `Delivery_date`. This will store the date that the goods are delivered to the customer.

When you have set that one up, create another field and call it `Delivery_time`. This will be an Integer field of three digits and it will be calculated. For the calculation you need to enter:

`Delivery_date - Order_date`

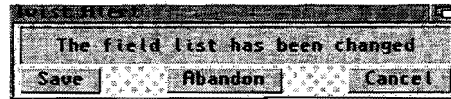
(It is this way round because delivery takes place after the order and therefore is a larger number). Don't forget to check that the attribute Not Editable is ticked.

When you have set this up, your field list should look something like this:



some fields have been added

To leave the screen, click on the top left hand corner. You will then see the message:



Click on **Save** to save the new structure and return to the List Mode. Now if you scroll the record list to the right, using the slider at the bottom of the screen, you will see the new fields.

Adding new fields after the database has been in use for a while is more restricted than creating them to begin with, because you cannot make the fields *required without* causing some confusion. They shouldn't be *validated* either, for the same reason, in that there are already records existing that will be invalid because they are blank in that field.

Editing Fields

Before we go any further with changing the structure of the database, you need to edit each of the records to enter a date into the `Order_date` and `Delivery_date` fields. Try to vary them so that some were shorter delivery times than others. So, on the record screen, making sure that the cursor is on the first record, select **Edit**. Enter the dates in the `Order_date` and `Delivery_date` fields and select **Save**. Move the cursor down to the next record and repeat. Continue until all the records have been updated.

Now that is done, we can edit the `Order_date` field to make it **Required**. We shouldn't do that with the `Delivery_date`, because at the time when a new record is added, a customer has placed a new order and the `Delivery_date` may not be known.

Before we can make this next change though, the boss comes in again! He has had an idea about analysing sales according to geographic area and wants to change all the customer codes! They will now be pre-fixed by one or two letters, e.g. SE for London and the South-East of England, N for Scotland etc.

Our `Customer_no` field is an integer and it won't accept characters, so we'll have to change that as well.

From the **Project** menu select **Edit fields**. This gives us the *Edit fields* requester as before. The line cursor should already be on `Customer_number`, as it is the first field in the list.

Select **Edit** and you will see the *Field definition* requester, with the details of the field being edited; note that you cannot now change the name of this field. To make the required change, check **Text**. The field length was 5 digits, of which only 3 were used, but the extra length was deliberate to allow for the growth of the company. So it still needs to be larger than the largest current customer number which will be 2 characters and 3 digits. However, using the letter prefix, allows us to re-use the same numbers, so 4 digits should be enough. This makes the field 6 characters long.

When you have made that change, click on **OK**. This will take you back to the *Edit fields* requester. Move the line cursor down to the field **Order_date**. Select **Edit**. Turn on the attribute **Required** by clicking on it. Finally, click on **OK**.

The two fields have been changed, so now click on top left hand corner of the screen; select **Save**.

Once you have completed the field editing procedures, you will have to edit the records again to change the customer numbers. Change 100 to SE100, 200 to SW200, 300 to NW300 (you shouldn't have any of these) and 400 to N400. In the next part of the tutorial, you will see how to obtain a list of sales from just one area.

Deleting Fields

Having learnt how to add extra fields and how to edit fields, the only operation we have not yet encountered is the deletion of fields.

It is unusual to want to delete fields from a database but, if the file is large, with a many records, you may well start to be concerned about disk space and want to remove any unnecessary fields. There are two things to look for. Either a field which doesn't often have data entered in it, or a field which doesn't appear on any reports. Both of these waste valuable disk space. Of course our database isn't large enough to worry about this, but we will go through the procedure anyway!

Select the **Project** menu, **Edit fields**. You will see the *Edit fields* requester, with the field list. The cursor should be on **Customer_number**. Click on **Delete**. The field disappears from the screen. *Don't Panic!* At this stage it is not too late to retrieve it. Click on the top left-hand corner of the screen and you will see the message:



Click on Cancel. You still have the field list and Customer_number is still deleted. Click on the top left-hand corner again and this time select Abandon. Now you are back at the record form and the field Customer_number is still there!

If you had selected Save, you would have saved the new file structure and when you looked at the record screen, Customer_number would have gone.

You may have realised that, although it is useful to record how the order is placed, this file is not the most appropriate place for it. Also, what happens when the order is placed by letter and then altered by phone? So you decide to save file space by deleting this field.

Select Edit fields again. Left-click on Order_type and click on Delete. The field disappears from the list. Now click on the top left-hand corner of the screen and select Save.

You will return to the record form and the field Order_type will have gone. It cannot now be retrieved.

Summary

In this section of the tutorial you have learnt:

- Why backups are important.
- How to change the record structure by:
 - Adding new records.
 - Editing existing records.
 - Deleting old records.
- How to change the database structure by:
 - Adding new fields.
 - Editing existing fields.
 - Deleting fields no longer required.

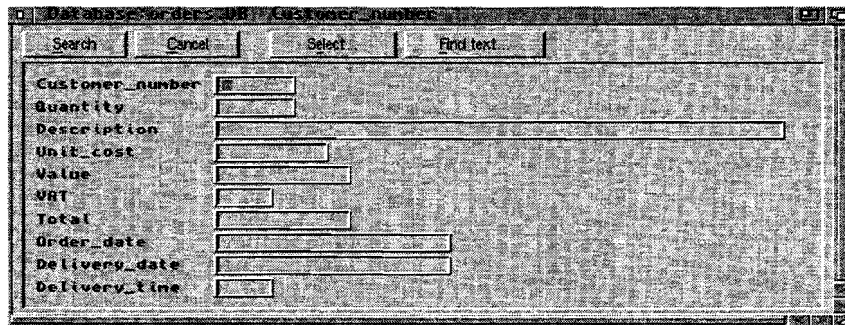
Tutorial 3 - Reporting

Now that we have some data in our database, we are ready to start getting meaningful information out again, in reports.

There are two main ways of obtaining information, the screen query and the formatted report. Reports can be saved, and used time and time again and can be saved to another file or printed, as well as being displayed on the screen. Screen queries need to be set up each time, but are quick to use and are only displayed on the screen. There is a certain amount of overlap between the two functions as you will see when we work through the examples.

Screen Queries

To set up a screen query, select **Query** from the record form. You will then see the query form which looks like this:



the Query form

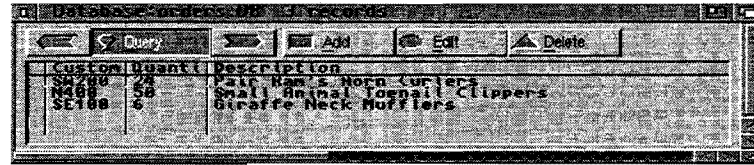
As you can see, there are four options, **Search**, **Cancel**, **Select** and **Find Text**.

Search

Search allows you to list all records that exactly match a particular item of data. First move the cursor to **Quantity** by pressing the down cursor key, **Tab** or using the mouse. Enter 20, then click on **Search**.

You will see the record screen, with just one record, the 20 Monkey Tyre Swings. Clicking on **Query** will restore the full record list.

Select Query again and this time when you see the query form, the number 20 is already in Quantity. Delete that and move the cursor down to VAT. Enter 0 and select Search. This time you should have a list of three records, all with zero VAT.



The screenshot shows a database window titled "Database: orders.DB 3 records". It has a menu bar with "Query", "Add", "Edit", and "Delete". Below the menu is a table with three columns: "Customer", "Quantity", and "Description". The table contains three records: "SA200 24 Hair Man's Worn Curlers", "N400 50 Small Animal Toenail Clippers", and "SE100 6 Giraffe Neck Mufflers".

Customer	Quantity	Description
SA200	24	Hair Man's Worn Curlers
N400	50	Small Animal Toenail Clippers
SE100	6	Giraffe Neck Mufflers

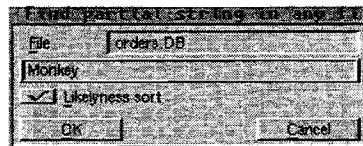
records with zero VAT

Click twice on the Query button. Delete the zero from VAT and move the cursor up to Description. Enter the word Monkey and click on Search. You see no records listed and yet you know you have at least one record with the word Monkey in Description!

To use Search, the record being searched for must be an exact match for what is entered in the template.

Find Text

Go back to the query form again. This time select Find text. You will see a box, entitled Find partial string in any field, with a box for the file, a box for the text, and a checkbox for Likelihood sort. Type Monkey into this box and select OK.



using Find text

You will see a list of one record for the Monkey Tyre Swings. Go back to the query screen again and select Find text. Delete Monkey and type in 0. What do you see this time? Can you see why?

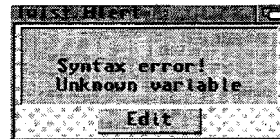
Return to the query screen again and select Find text. This time delete on and type in 50; select OK. What do you see? What did you expect? If you had typed in 0, you would have selected all of the records, because they all have a zero in the Customer_number, which is a text field, but 50 only appears once, in the Quantity, which is an integer field and therefore not searched in a find text operation.

*Find text allows you to search for all records containing a particular word, or part of a word and for digits contained in a text field.
It will not search numeric fields.*

Select

Go back to the query form again. This time choose **Select**; you will see the *Expression* requester

Enter `Description#Monkey` in the formula box (the Twist operator # means *like* and appears near the bottom of the list of operators). Click on OK to see whether this will work.



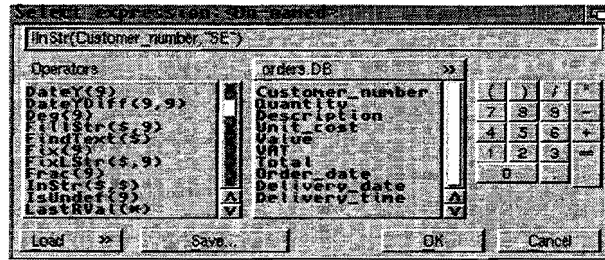
The above alert appears because you must enclose words in double-quotes (") when using the *Expression* requester; so select **Edit** and put double-quotes around the word **Monkey** and try again.

No records are listed; to search for a description using an expression it must be exact. The only difference is that using the *like* operator, instead of equals, means that the program ignores the difference between upper and lower case letters. So, try `Description#"monkey tyre swings"`.

Go back to the query screen again, this time we are going to list all records for customers in the South-East of England.

Delete any expression and, from the list of operators select **Not**, then **InStr**. **InStr** comes with a set of brackets, with two dollar signs inside. Replace the first dollar sign with the name of the field in which you are looking for the string, i.e. `Customer_number` and replace the second dollar sign with the string you are looking for, which must be in double quote marks, i.e. `"SE"`. Make sure you keep the comma between the dollar signs.

The requester should look like this:



using InStr

Instead of selecting OK, select Save. You will be prompted for a title, so type in South-East and select OK. This expression will be saved under the name South-East and, if you look at the top of the screen, where it says, Select Expression, you will see that this has changed from <unnamed> to South-East. Now click on OK and you will see the list of orders for customers in the South-East of England.

Go back to the query form; this time you have a choice to either use South-East or Edit expression.

Selecting South-East, will take you straight to the list of records for those customers. Instead, select Edit expression. When you see the expression definition requester, delete the expression using the Backspace key and select Load. A small box will appear, next to Load, listing all saved expressions. In this case it will only be South-East.

Click on that and the expression will re-appear in the formula box. Move the cursor to SE and delete it. Type in N and click on OK. You will now see the list of orders for customers in the North, but you have not altered the expression as it was saved. It would also be possible to go through the different geographic areas, to set up and save an expression for each one, in order to be able to obtain regular listings without re-typing the formula each time.

Select allows you to select records according to a formula, or calculation and to save the expression to be re-used another time.

Creating Formatted Reports

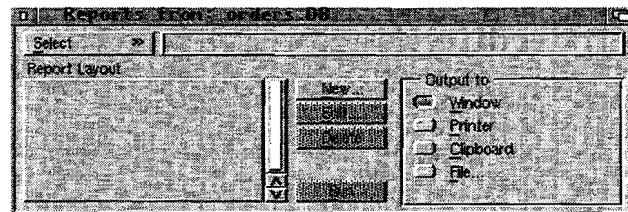
A database is a useful way of storing and retrieving information. However the power of a good database is enhanced by the ability to display selected fields of that information in different ways for certain applications.

For example, you have created a database of customer orders. This database contains name and address details along with an itemised list of goods ordered, charges made for those goods and their despatch dates.

Your customer phones up enquiring where some of the goods are because only half of them have arrived. Thus, you want to search on the name and address fields, but when the record is found it is not necessary to display the address or payments again, merely to list the name and the goods ordered with their despatch status.

Setting up a Report

To set up a report, select **Edit report** from the **Record** menu. You will then see the *Report* form which looks like this:



setting up a Report

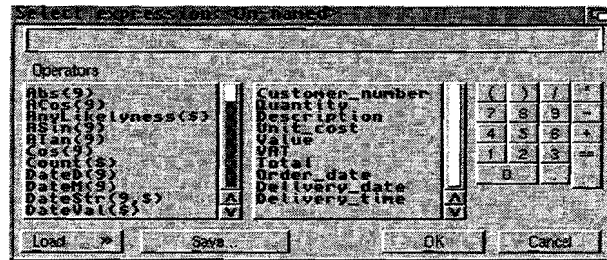
Report Selection Criteria

By using **Select** we can enter criteria that will be used to decide which records will appear on the report.

If the **Select** field is left blank (as you see it in the form above), all the records will appear in our report. Alternatively we can type in a formula, if we know exactly what we want or we can left-click on **Select**...

Click on **Select** to get a pop-up, listing any expressions we have already set up and saved (in this case **South-East**) or **Edit expression**. So, once an expression has been set up using a screen query and checked to be certain that it is producing the right records list, it can later be used within a formatted report.

In this case, however, we are going to set up a different report so select **Edit Expression** from the pop-up. We obtain the familiar *Expression definition* requester:



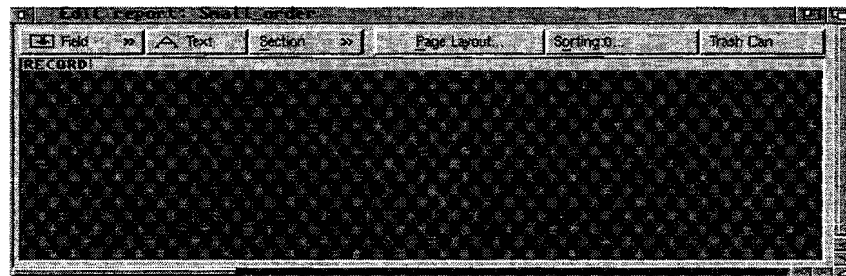
selecting particular records for the report

Let's say that we want to set up a report that lists all orders where the value of the order, before VAT is added, is one hundred pounds or less, because these orders cost us comparatively more in administrative costs against profit. So the formula we want is Value less than or equal to 100.

Select Value from the field list, less or equal from near the bottom of the operators list and type in 100, or select it from the numeric keypad. Choose Save.

For the expression title enter Small_order; then click on OK. This expression has now been saved and so can later be accessed from Query if you wish. Click OK on the *Select expression* requester.

From the *Report* form, select New; a pop-up will appear. Choose Empty report and, when the file selector appears, give the report the name Order.Small_order.R; you can only change the middle part of the name, that between the two full-stops or periods. Select OK. You will now see the *Edit report* requester:

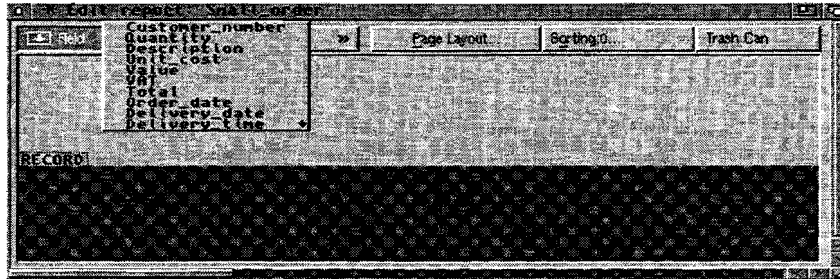


preparing to lay out a report

Including Fields in a Report

In the top left-hand corner above the shaded area is the word **RECORD**. Click on this, hold the mouse button down and drag downwards. The white area that grows with this is the amount of space allocated to each record in the report and in this area you will be arranging the fields you want to include, in the order you want them displayed. Let go when half the window is white and half is shaded; you can change this at any time

To include a field in the report, click on **Field**, in the top left of the screen. The field list appears.



choosing a field

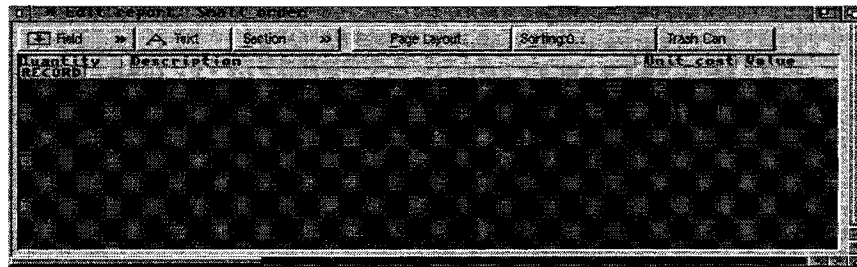
Move the cursor to **Quantity**, click the mouse button, hold and drag down. The field list disappears and, when you drag on to the report area, you will see a box. Move the box to a position in the report area where you wish the information to appear and let go the button.; the word **Quantity** appears in the box.

This box represents the field **Quantity**. Click on the box once more and a border appears around it, with a small square in the bottom right-hand corner. Click on this square, hold and drag to the right; the box gets larger. Now move it back to the left, the box gets smaller. Adjust it to the same size as the word inside it.

Now click on the box (not the square) and keep the mouse button pressed down; the box goes white. Drag the box to the bottom of the white area and over to the right. Let go of the mouse button. The field has moved. If you attempt to drag the field into the grey area, it will jump back to its previous position, once you release the mouse button. Now move this **Quantity** box back to the top left hand corner of the record area.

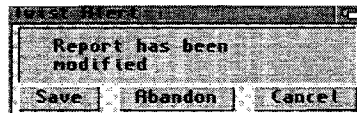
Now you know how to adjust the field size and how to move the fields to where you want them. Select **Field** and select **Description** from the field list. Place this to the right of **Quantity** in a straight line. This may take up more than the width of the screen, if so, scroll across to the right using the horizontal slider at the bottom of the screen. Then select **Unit_cost** and **Value** and place them in the same line.

Finally, scroll back to the left and click on the word **RECORD** again, hold and drag it back up to the bottom of **Quantity**. If you leave it immediately underneath the field, there will be one line between each record in the report, or you can place it on top of the field, in which case there will be no blank lines in the report. Your screen should now look like this:

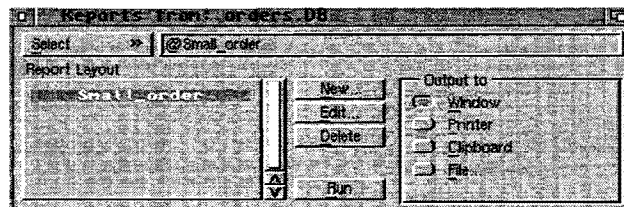


a simple report

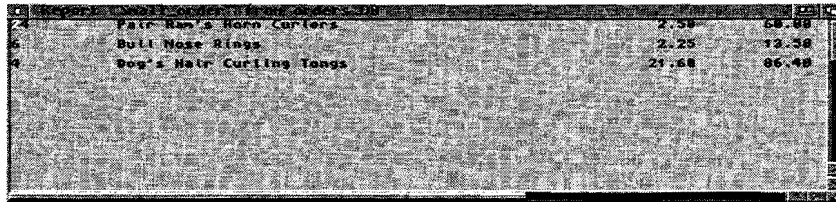
Click on the top left-hand corner of the screen. You will see the message:



Select **Save**. You will return to the report form. **Small_order** is now listed under **Report Layout**, but the **Select** box is empty. Click on **Select**. Select **Small_order** from the expression list. Check that **Output to** is directed to **Window** and that the form looks like this, with **Small_order** selected:



Then select **Run**. You should see the report on the screen, listing three records. You can scroll up and down the report and from left to right, using the sliders at the right-hand side and the bottom of the screen.



2	Pair Ram's Horn Curlers	2.58	68.88
4	Bull Nose Rings	2.25	13.50
4	Dog's Matr Curling Tonga	21.68	86.40

the finished report on screen

When you have finished looking at the report, click on the top left-hand corner of the window to return to the report form.

Once you have a number of expressions set up and saved and a number of report layouts, it is possible to 'mix and match' using each expression with each layout in turn.

Editing Report Layouts

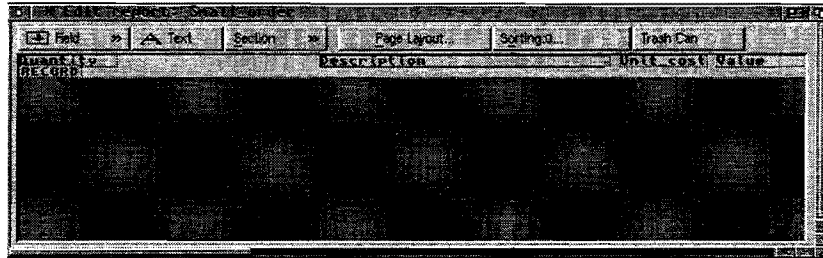
We are now going to edit our report layout, to make it more informative. With the cursor on **Small-order**, select **Edit**; we see the report definition window again containing our current layout.

When you ran the report and looked at the output before, there was a good deal of space between the end of the description and the unit cost. This is because the description field is 50 characters long, but most of the existing descriptions are shorter than that. So perhaps we can shorten that field a little.

Click on the box labelled **Description**, to bring up a bold border. You may have to scroll the screen to the right to find the small square that becomes visible, at the bottom right of the box. Click and hold on the small square in the corner and drag about 4 cm to the left; let go the mouse button.

Then click on and drag the whole field to the right, so that it is still next to **Unit_cost**. This creates extra space on the left hand side of the screen, which is useful because we are going to add the **Customer_no** field, so that we can see if a particular customer is always ordering small quantities.

So far you should have achieved something like this:



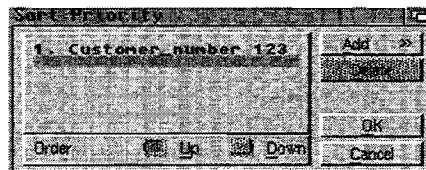
Now move **Quantity** to the right, so that it is next to **Description** again. Then select **Field**, choose **Customer_number** from the pop-up (select, hold the mouse button down and drag) and place this at the left-hand side of the record; then expand it to fill the space.

Close the report by clicking in the close box, choose **Save** and try running the report again; notice the difference

Sort Orders

Long reports containing hundreds of records are very difficult to understand, so to make it easier we often group the records according to some system, which will vary depending on the function of the report. We call this the *sort order*.

In order to be better able to see which customers make the most small orders, we need to sort the report into customer number order. So choose **Edit** again and, when the *Edit report* window appears, select **Sorting** from the top right of the screen. You will see the *Sort Priority* requester.

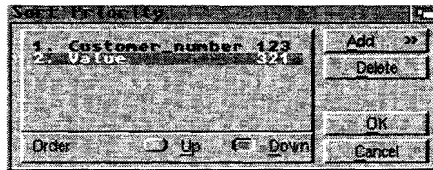


defining a sort priority

Select **Add** and, from the field list pop-up, choose **Customer_number**, then make sure that **Up** is lit.

We have now selected the first sort field, but we haven't finished with the *Sort Priority* requester yet. If we had a database with hundreds of records in it, just sorting by customer number might not be adequate to make sense of the report, within each customer group we may want to sort further.

For example, we could choose to sort the descriptions into alphabetical order, or sort by quantity but, as this report concerns the value of the order, it makes more sense to sort on the value. So select **Add** again, select **Value** from the field list pop-up and this time make it a descending sort order by choosing **Down**. (you will probably need to click on **Value** in the window to ensure that it is selected before choosing the sort order).



The cursor has to be on the sort field you are changing.

Select **OK**, you will return to the *Edit report* window, which will look just the same, even though you have added a sort order.

Adding Text

To create a well-presented report, you need to add text to the layout. First of all, the report should have a title, explaining what it is about. It would be useful as well to have headings for each column, for the same reason.

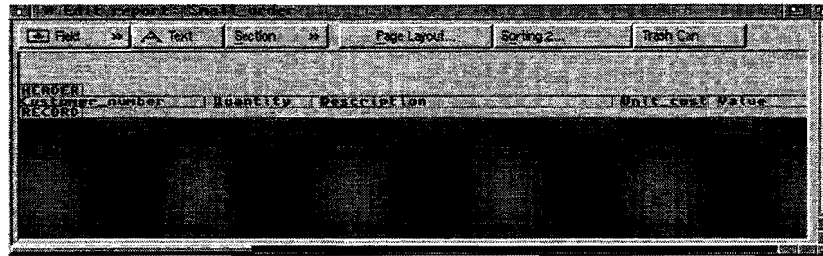
To add these we first need to create a new section of the report, because, if we put this text in the record section, it will be repeated for each record.

Adding Report Sections

Select **Section** from the top of the window. A list of optional sections of the report will appear. There are two that would suit our purposes, one is **Header** and the other is **Before Report**.

A *header* is repeated at the top of every page, whilst text *before report* appears once only, just before the main body of the report. For a report containing only three records, this will obviously amount to the same thing, but not many reports will be quite that small, so we are actually going to use **Header**, because we would want the headings to appear on every page if the report was longer.

Select Header. The header section appears above the record section. You will need several lines for this section, so click and drag down on the word HEADER.



adding a header

Click and hold on Text at the top of the window and drag the text box to the middle of the screen on the top line of the header section. then double-click on the text box to bring up the text requester. Backspace four times to delete Text and type in:

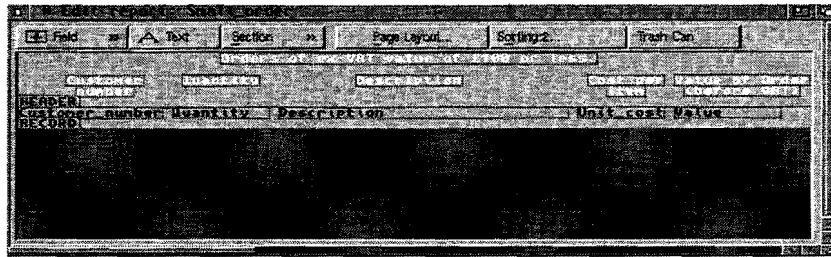
Orders of ex-VAT value of £100.00 or less.

Select OK. The title of the report is now on the top line and will appear at the top of each page of the report. If you made any typing mistakes, double-clicking on the box will bring the requester back to the screen so that you can edit it. You can position the text by dragging it around on the screen as you wish.

Below the level of this text box, you now need to enter some more text boxes. Above the field Customer_number, place two boxes, one above the other. In the top one type Customer and in the bottom one type number. Then, working across the report, put one box above Quantity, enter Quantity. Above Description put one text box saying Description of Goods then two boxes above Unit_cost, the top one saying Cost per and the bottom one item.

Finally place two boxes above Value, type Value of Order in the top one and (before VAT) in the bottom one. By having all the text in separate boxes, it is easy for you to move them around so that each column heading is centred above the field it refers to.

You now have four lines in the header and it should look like this:



the completed report

Click on the top left-hand corner of the screen and select **Save**.

Running the Report

You have previously run the report, directing the output to Window. This time select **Printer**. Check that your printer is on line and loaded with paper etc., then click on **Run**. The report will print, complete with the headings you just set up. If you do not have a printer connected, you can choose **Window** or **File** as you wish.

After the report has run, change the output setting back to **Window** and click on the top left-hand corner of the *Reports* form. You will return to the record window .

Select the **Report** menu (right-click) and you will now see that, in addition to the **Edit report** option, the report **Small_order** is listed. Select this to run the report straightaway, with output directed to the screen. As you can see, this is a shortcut. If you do wish to edit an existing report, or set up a new one, select **Edit report**.

Installing an Icon

There is another shortcut to running a report. For this you need, first of all, to leave the orders database file and return to the *Twist* program screen. To do this click on the icon in the top left-hand corner of the window.

From the Twist program screen, select the Project menu and choose Install icon. From the file selector, select orders.Small_order.R Then select OK. You will then see the Install Report Icon requester, with a Report icon in the frame. Select OK and the report icon will appear on the Twist Icons window, with the title Small_orders underneath it. This can then be moved to the position you want within the window. You can, of course, assign a keyboard shortcut to this icon.

In future, to run the report, all you have to do is double-click on this icon from the Twist Icons window, without needing to open the database file first; you are prompted to Perform the report, select OK. Once the report has run, the output will appear on the screen. To print it, drag the Small_order icon onto the Printer icon within the window, delete it by dragging to the Trash; dragging to the Clipboard will be covered later.

Related Reports - Linking Two Files

Once a report is set up, it is sometimes useful to be able to bring in information from a second file and show it on the same report. In this instance, let us suppose that, in order to put the small orders report to its best use, we want to be able to contact the customers who have placed these orders, to try and persuade them to increase the value, or to tell them that we are charging an extra handling fee.

It would be useful to have a telephone number and a contact name against each customer, to make this easier. To do this, we would need a database file with this information in it.

Normally we could expect to have a customer file already set up, with name of company, contact name, address, telephone number and probably other data too. So, to revise what we learnt in Tutorial 1, we will now set one up.

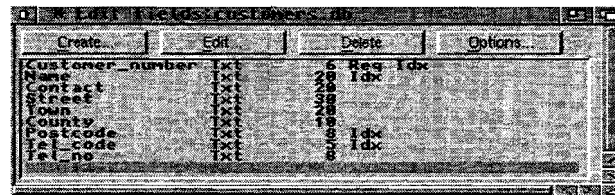
Remember to start from the Twist program screen. Select the Project menu and New, At the file selector, type in customers.DB and select OK.

Create the following fields:

Customer_number	Text	6 characters	Required	Indexed
Name	Text	20 characters		Indexed
Contact	Text	20 characters		
Street	Text	30 characters		
Town	Text	20 characters		
County	Text	10 characters		
Postcode	Text	8 characters		Indexed
Tel_code	Text	4 characters		Indexed
Tel_no	Text	8 characters		

Note: It is essential that the field `Customer_no` is indexed. Files can only be related in a report if the field being used in the second file is indexed.

The fields are all text, even though some of them contain digits, because they are not numbers in the mathematical sense i.e. we will not want to use them for calculations. Click in the top-left of the window to exit the requester; your list should look like this:



a customer details database

Once you have done that, leave the *Edit fields* requester, by clicking on the top left-hand corner. Add the following records, remembering to Save after each one.

```
SE100
London Zoo
Elly Fant
Regents Park
London

W1
0171
8374021

SW200
Soames & Partners
Jack Hare
```

12 Little Lane
Norton Fitzwarren
Somerset
TA12 6QT
01356
715826

NW300
Pampered Pets
Tom Katz
11 High Street
Clevelys
Lancs
BL24 9RS
01891
654876

N400
The Vetinary Surgery
Peter Scott
81 Upper Street
Dalkeith
Lothian
EH35 9PT
01564
431986

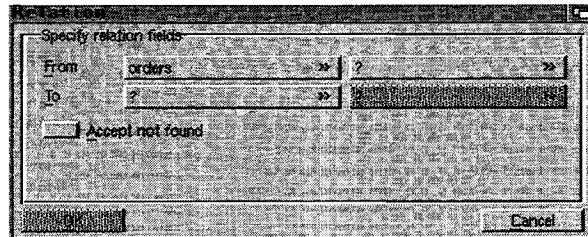
Select **Cancel**. That is the second database you have set up and entered records for; we hope it was easy to remember how.

Now leave this database and return to the Twist program screen, by clicking on the top left-hand corner. Open **orders.DB** and select **Edit report** on the **Report** menu. With the cursor on **Small_orders** select **Edit**. We are now going to combine data from both our databases within this report.

We only need the customer information once for each customer, so we are going to add a new section to our report. Select **Section, Before 1st Group**. This will bring up the field list; choose **Customer_no**.

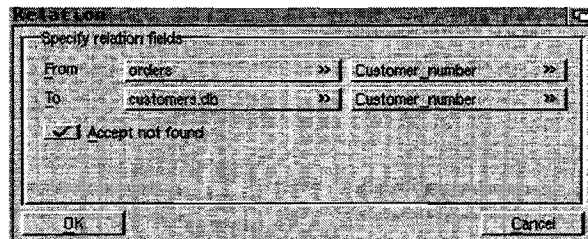
The report is already sorted into customer number order, so now we will have a heading for each group, i.e. for each customer number. The new section will appear between the header and the record sections. You need to click and hold on the **BEFORE** and drag downwards to make more space.

Now select **Field** from the top-left of the screen; at the bottom of the field list (move the mouse down to extend the list), you will see **<Change file>**. Click on this and then select and click on **<New file>** when another pop-up appears and you will see a requester:



Click on the question mark under **orders** in the top row and to the right of **To**. This will bring up a list of other database files in the current drawer. In this case, there should only be **customers.DB**. Select this and the field **Customer_number** will appear in both rows. This is because we set up that field in each file with the same name and structure, so the program can match them automatically. Otherwise we would have had to select from field lists obtained by clicking on the question marks in both the boxes.

Note: The field selected for the **To** file must be indexed.



the completed Relation requester

Make sure that **Accept not found** is on, that is, that there is a tick in the box. This means that if a record is found in the **orders** file that doesn't have a corresponding record in the **customers** file, it will still be included in the report, but the customer information will be blank. This is important, because you want to know if you have got orders for non-existent customers!

Select **OK**. Now if you call up the field list (select **Field**), you will find that it is a list of the fields in the customer file.

Select Name and place it about 4 cm from the left of the Before 1st Group section, at the top. Notice the difference in the way the field is labelled. Then select Contact and place that immediately underneath. Place a text box to the left of each of these fields. In the top one enter Company name and in the bottom one enter Contact name.

To the right of the field Contact, place another text box and enter Tel no: then, to the right of that text, place the two fields Tel_code and Tel_no, in that order. Adjust the depth of the BEFORE section, if necessary, to give three lines; check that your screen looks something like this:

the completed report

Now click on the top left-hand corner of the screen. Select Save and return to the Reports form.

Delete @Small_order from the Select box at the top of the screen, to leave it blank so that you have the whole file listed to the report and choose RUN. You will now see that the records are sorted by customer and, before each customer group, is the relevant information for that customer.

Click on the top right hand corner to remove the report from the screen and at the Reports form, restore the selection to Small_order, run it again and you should see the following:

Report: Small_order: Items orders: 00

Orders of ex-VAT value of £100 or less.

Customer number	Quantity	Description	Cost per item	Value of Order (before VAT)
Company name: Peter Scott				
Contact name: The Veterinary Surgery		Tel no: 01564 431966		
N488	4	Dog's Hair Curling Tongue	21.50	86.00
Company name: James & Partners				
Contact name: Jack Hare		Tel no: 01356 715626		
SM290	24	Pair Ram's Horn Curlers	2.90	69.60
SM290	6	Bull Nose Rings	2.25	13.50

powerful Twist reporting

Summary

In this section of the tutorial you have learnt:

- How to select records for a report.
- How to define a report layout.
- How to sort the records in a report.
- How to add report, page and section headers to a report.
- How to run a report from the record screen.
- How to install an icon and to run the report from the *Twist Icons* window.
- How to combine data from two files into one report.

You have also revised how to create a database file and enter records.

Twist Menus and Buttons

Most Twist functions can be accessed through menus or by buttons embedded in the top border of a window. What appears on the menus and buttons depends on whether the *Twist Icons* window is active or a database window is active.

The Menu

The Project Menu

The number of items on the **Project** menu will vary depending on what you are doing with Twist. When you first enter Twist with no database loaded, the menu will look like this:



If you are working with a database, report etc. there will be additional items relating to editing fields, managing indexes and relations:



These extra entries will be described later under the relevant sections; the common items are detailed now.

New

Use New to start work on a new database; a file selector will appear from which you can choose the path and filename of the database that you wish to create. The file extension for a database file must be **.DB**; Twist will append this file extension if you do not specify it yourself.

Once you have chosen a filename for your new database, the *Edit fields* requester will appear and you can start creating fields for your data.

Open...



Open allows you to open an existing database file, a report, a text file or to run a script or ARexx program; how Twist treats a file depends on the type of the file and its final file extension.

The list of extensions which Twist understands and their use is:

.DB	Database files
.S, .rex	Script or ARexx files
.txt	Text file
.R	Reports
.I	Import definitions
.E	Export definitions
.U	Update definitions
.M	Mailmerge definitions

You will open a database file in order to work with it or to make changes to the file. Again, the default (and mandatory) file extension is **.DB**. The file will be opened when you select OK and it will be displayed in either List mode or Record mode, depending on which mode was current when the file was saved.

If you open a file with an extension of **.S** or **.rex**, the script or ARexx program will be executed and then control will be returned to Twist.

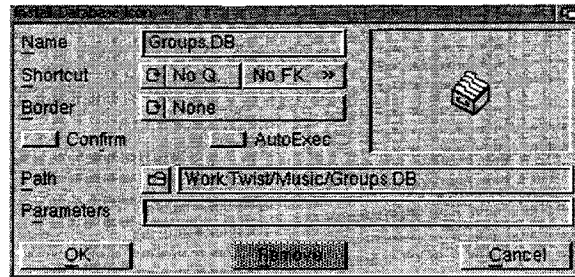
Opening a report file (**.R**) will run the report.

Install icon...



Use this to install an icon in the *Twist Icons* window - you will then be able to manage your databases, reports etc. from within this window by double-clicking on the icons (or using the designated keyboard shortcut). You can also delete icons using this command.

The first thing you see is a file requester - choose the file that you wish to install and select OK; the *Install Database Icon* requester will appear.



The icon that is selected and appears within this requester will be dependant on the file extension of the chosen file. The default icons that Twist uses are contained within the Icons drawer and there is an icon for all the common filetypes plus icons that are to be used in requesters to designate pop-ups, an icon for the Trash can icon etc. You can design your own icon and include it within this drawer (giving it the correct name), if you wish; it is best to edit an existing icon (with IconEdit for example) rather than to design one from scratch, to ensure the correct size.

Within the *Install Database Icon* requester, you can select the name that will be attached to the icon (which can be different from the filename), assign a keyboard shortcut to the icon, choose a border for it and select various other options described below.

Shortcut

You can choose a to assign a function key shortcut to an icon by clicking in No FK and choosing from F1 to F10; leaving No FK selected means no keyboard shortcut for this icon. To allow for more shortcuts than 10 (F1-F10) you can also choose a qualifier to go with the function key; select from No Q. (no qualifier), Amiga (either *Amiga* key), Shift, Alt or Ctrl.

Border

You may want a border around the icon, the icon and its name or no border at all; choose one of these from the Border cycle gadget.

Confirm

This option, if selected, causes confirmation of the action associated with the icon to be requested prior to performing the operation; this may be helpful if the operation is a long one such as a lengthy report to a printer.

AutoExec

If this is selected for an icon then Twist will automatically open the database, report etc. that is associated with the icon when Twist is started. This is useful if you have designed a database to be used by someone else, for example.

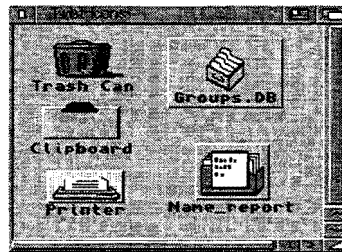
Path

Locate the file to be assigned to the icon using the Path file selector.

Parameters

This is a query string if the file is a database - only those records matching the query will be displayed. For a program it is the command line which is passed to the program.

The Twist Icons Window



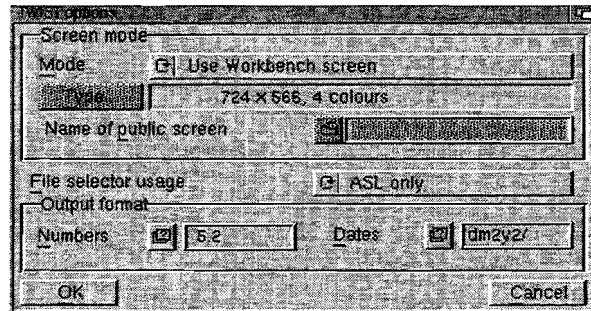
This is the window in which you can place icons associated with database files, reports, mailmerge files, etc. and then access these files quickly and easily by double-clicking on the icon.

To place an icon in the window, initially, use **Install Icon...** from the **Project** menu, as described above. If you select an icon from the *Twist Icons* window and then choose **Install Icon...**, you will be able to edit that icon's attributes or even remove it from the window.

Note that dragging an icon to the Trashcan will *delete the file associated with the icon*, dragging to the Clipboard will copy the file (if possible) to the Amiga's clipboard and dragging an icon to the Printer will print that file. again if this is feasible.

Closing the *Twist Icons* window will quit Twist.

Global options...



This requester allows you to customise Twist to your working environment; you can choose the screen on which you wish Twist to open, the type of file selector that Twist will use and the display format for real numbers and dates.

Screen Mode

Using the cycle gadget, choose a screen for Twist: **Use Workbench screen** will open Twist on the standard Workbench screen as you are using it; **Use public screen** allows you to name a screen that has been made public by some other application (enter the name of this screen in the box or choose it from the pop-up list); **Clone Workbench screen** will open Twist on new screen of the same type as that used by Workbench - this might be useful if your Workbench screen is rather cluttered but it will use more memory; finally you can select **Create own screen** where you can choose (by clicking on **Type**) from a pop-up list of all the installed screen types.

The most economical choice, from a memory point of view, is **Use Workbench screen** but there may be times (for complex databases, perhaps) where you will want a different screen - the choice is yours.

File selector usage

This allows you to nominate which of the installed requester libraries you will use within Twist. The choices are:

ASL only (supplied with Workbench 2.04 and above); Twist will then use this library for all file, font and screen requesters.

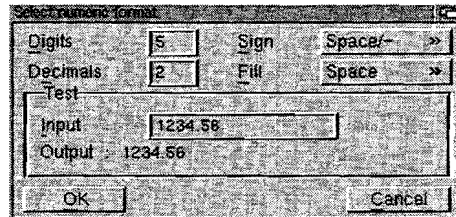
RCT Requester (supplied with Twist); this supports keyboard shortcuts in all requesters (including file, font and screen requesters).

ASL Requester; choosing this will use the ASL library for file, font and screen requesters and the RCT library for all other requesters (thus giving you keyboard shortcuts).

If you wish to use the last choice you must have both ASL and RCT libraries installed in your system.

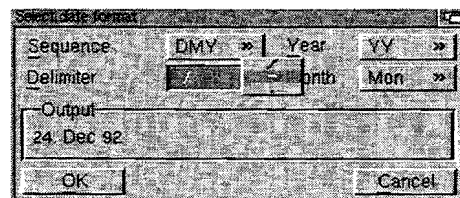
Output format

Choose the format of numbers and dates that you will use in this session with Twist. Click on the gadget next to the name to pop-up a requester from which you can select the required format.



The *Select numeric format* requester allows you to choose the number of digits before and after the decimal point, how to display the sign of the number and how to pad out the number if it shorter than the chosen format.

Try experimenting with the options and typing in an **Input** string to see how all this works (the result is shown after **Output**). Choosing **OK** will use this format from now on, **Cancel** aborts leaving the choices as they were when you entered this requester.



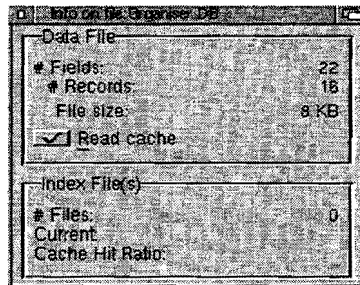
The *Select date format* requester lets you choose the sequence of day, month, year, the delimiter to use between these and the format of the year and month. Again you can experiment and view how the output changes.

Info...

This brings up an information box which shows the version number of Twist, its ARexx port name and your registered name and serial number. Please quote this information when asking for technical support.

Note that, although you can edit your user name and serial number, the changes will not be saved when you leave the box because **Install** is greyed and not available.

If a database is loaded and you select **Info...**, you will see some data about the database, including its size, number of records/fields etc.



File size

The value shown here represents the physical size of the database file on the disk drive. However, deleting records from the database does not automatically reduce this. When a record is deleted its position is marked as being available for use by another record. When adding a new record, the database is searched first for free spaces before creating a completely new record at the end of the file. If free space is found, then the contents of this record are over-written by the new data.

If you have deleted a large number of records from a database, or space is tight on the disk drive, then you may wish to reorganise the file to physically reduce its size and remove all the free positions within it. Twist will always re-organise a database whenever its field layout is changed. Therefore, select the **Edit fields** option from the **Project** menu and make a dummy change to the first field (press **Return**, select **OK** and close the window); then select **Save** to effect a re-organisation.

Read Cache

Twist operates a software device known as a *read cache*. The control on this requester allow you to turn the cache on or off. Normally, Twist reads and writes all of its database files to and from the disk drive. This is usually a fairly swift process and will not hamper the speed of Twist. However, when a database starts to become large, perhaps with a many fields in each record, then editing or searching the database can start to slow down. This is because each record has to be read from the drive, in turn, its data examined to see if it meets certain criteria and then copied into the report file (also on disk) if required. This can cause much disk accessing and may slow Twist down because access to a disk drive is not as efficient as access to the computer's memory.

If the read cache is switched on, the performance of Twist can be enhanced dramatically because a copy of the database file is always kept in the computer's memory. This copy is used in preference to the disk drive version since the computer's memory is faster and more efficient, ensuring that Twist can always run at maximum speed. Whenever a record is edited or added to the database, then its data is copied into the allocated memory block and *then* written to the disk. This ensures that both copies of the file match one another exactly.

The drawback with the cache is that it requires sufficient free memory that it can hold a copy of each open database along with the program. With a large database or with a number of databases in the machine at any given time, it may not be possible keep the cache open due to insufficient memory.

When a file is created the default cache state is on. If you don't wish to use the memory needed for the cache, simply turn it off. Twist will automatically free the memory block and save the new setting in the database file.

The current index File

We have already discussed the database cache so you should know what it is and why it is useful. Twist keeps another small area of memory set aside for another cache. This is reserved exclusively for use with index files. The minimum and maximum size of the index cache is defined in the `Twist.inf` file (under the `INDEXCACHE` heading) and these can be changed; do not do so unless you are confident that you need to change it.

Index files may be large and a number of them may exist for one database. For this reason the cache may only hold a few records from each index file at any given time and therefore can't always guarantee to be holding the required information. To keep up to date it will periodically have to fetch new data. The information shown at the bottom of the display, the **Cache Hit Ratio**, provides an indication of how successful it has been at directly providing records required for use, without having to fetch them from disk.

To understand how a small cache can be effective, even with a large database, it is necessary to understand about the internal workings of index files. Like the .DB file, the index files store information in groups known as blocks. An index block may contain 50 or so record entries. The index blocks are stored like the trunk and branches of a tree. To find a certain record Twist travels up the trunk and starts to follow a certain branch. This means that the blocks closest to the trunk are accessed much more frequently than those closest to the leaves. The index cache holds the blocks which have been most recently and frequently used.

The value shown next to **Cache Hit Ratio** represents the effectiveness of the index file cache. The percentage value displayed represents the relative number of times a read from the disk could be avoided because the required block was already held in the cache. Please bear in mind that this only shows how effective the cache for the current field has been so far. The broader view of the total performance of all index files must be considered, to gain a true idea of how well the cache is improving the performance of Twist.

Click the close box (top left) to exit the information requester.

Quit

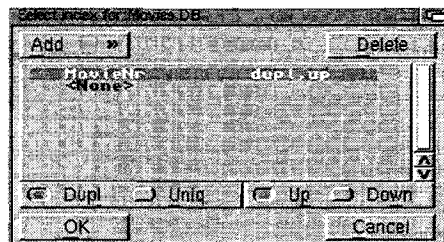


Quit is used to quit Twist; all windows are closed and if there is any unsaved work you will be asked for confirmation prior to any work being lost. **Quit** is equivalent to closing the *Twist Icons* window.

Index...

AK

The Index... option allows the indexes associated with a database to be manipulated:



The active index may be set by selecting it (or <None>), together with whether it is used in ascending or descending order and whether the index should be unique, or with duplicates allowed.

A new index may be created by selecting one from the Add pop-up, or an existing one may be deleted by selecting it and using the Delete button.

More detail on the creation and use of an index is given in the *Index* subsection of the *Creating a Database* chapter.

Relations

When working with relations in a form, the menu options under the Relations sub-heading are available. In general these commands operate based on the position of the input cursor, hence you should move the cursor to the field on which you wish to base the operation, prior to invoking the command:

Find record

F1

Find record is used to find a record in an N:1 relation where you are unsure of the value for a link field; for example: in a text field, enter the search expression his* and select Find record, or press F1. Twist will use the contents of the field as a search expression to select records from the related file. Depending on the number of matching records found, Twist will act as follows:

0 records:

Twist asks if you wish to create a new record:

The manner in which this occurs depends on the setting of the Allow create/edit in form flag, if not set Twist will open the related file's window and copy the field's value into the new window; otherwise Twist will create the new record in the current window.

1 record:

The record is simply displayed.

2 or more records:

Twist displays an alert indicating the number of records found.

If Show list is selected then the related file will be opened in list mode. The required record may then be selected by moving the block cursor to the record and pressing F2 (Reverse link):

Reverse link

F2

Reverse link does the opposite of Find record, moving instead from the child database to the parent database; it is normally used in conjunction with Find record.

Empty fields

F3

Empty fields is used to "unlink" a related record; this is achieved by moving the cursor to the field of the file which you wish to unlink and selecting the command. Twist will empty the fields for that file (and its subfiles). If the relation is a 1:N relation, confirmation will be required first since the dependent records will automatically be deleted.

Open window...

F4

This command opens the related file's own database window and shows the current record. This can be useful where a related form does not show all of the field in the file and some additional information must be entered into the related file.

The Record Menu

List mode

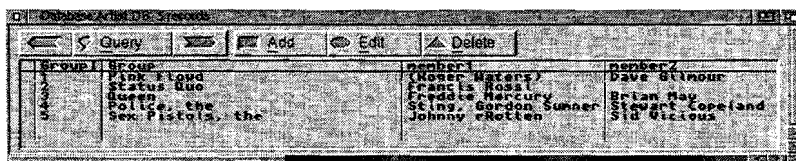
AT

Record mode

AR

Records within Twist can be displayed in two ways; *Record mode* or *List mode*. Displaying data using Record mode means that one record is displayed on the screen at a time, using either the standard form, or your custom one.

When there are several records in the database, it can be more convenient to display them using the list mode, where records are displayed in a list on the screen. To do this you need to right-click on Record on the menu and select List mode. To return to record mode, right-click on Record, then select Record mode. List mode looks something like this:



Group	Group	Member	Member
1	Link Stank	Robert	David Stank
2	Status Quo	Francis Rossi	
3	Queen	Freddie Mercury	Brian May
4	Police, the	Sting, Gordon Sumner	Stewart Copeland
5	Sex Pistols, the	Johnny Rotten	Sid Vicious

To browse through the records in record mode, click on the arrow icons either side of Query, or use the left/right or up/down cursor control keys. A single click on the arrow icons will move you through the database one record at a time. Holding the Shift key down while clicking the arrow icons will take to the start or to the end record of the database, as will holding the Shift key down while pressing up-arrow or down-arrow. Please note that if the database is quite large, there may be a short pause when moving from one end to the other.

If there are more fields in a record than will fit on the screen, use the up/down slider at the right-hand side of the screen to show the other fields, or enlarge the window.

In List mode, the arrow icons again allow you to browse through the records, as do either the left/right or the up/down cursor control keys. If there are more records than will fit in one window, then you can also use the up/down slider at the right-hand side of the screen. If there are more fields in each record than will fit on the window use the left/right slider at the bottom of the screen to move along the fields.

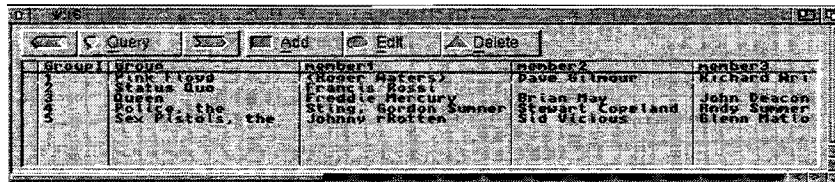
Configuring the List mode display

By default the List mode shows the database fields across the screen in the sequence defined when creating the field layout. Initially, each field will have the width and font defined by its default output format or its length; each field will be separated by a thick grey line.

Column order and width

The order of the columns in the list mode and the width of each column may be edited very easily by clicking and dragging the mouse.

For example, to change the width of a field move the mouse pointer over to the far right hand side of a heading, press and hold the mouse button down; the heading will be surrounded by an outline box which will move with the mouse.



Group	member1	member2	member3	member4
1	Pink Floyd	Robert Waters	Dave Gilmore	Richard Wright
2	Status Quo	Francis Rossi		
3	Queen	Freddie Mercury	Brian May	John Deacon
4	Police	Sting, Gordon Sumner	Stewart Copeland	Noddy Sumner
5	Sex Pistols, the	Johnny Rotten	Sid Vicious	Glenn Matto

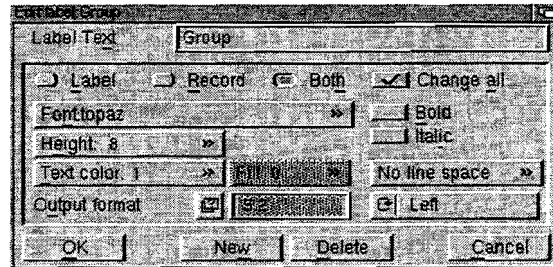
Adjusting the width of a field

While holding the mouse button down during this drag operation, the width of the field will be displayed in the title bar of the window. This value represents the width of the field using the current character and font settings.

To change the order in which the fields appear on the screen, click in the middle of the label and hold the mouse button down. It will now be possible to drag the selected field around and let go of it in a different position. If the mouse button is let go on top of another field, then this field will be shuffled over to the left or right by one position and will be replaced by the field which was dropped into the column.

Editing column appearance

Many aspects of the appearance of heading and its columns can be edited, including the font and font size. To alter a column's attributes, simply double-click on the heading of the required column. The resulting requester will look something like this:



Twist always saves changes to the List mode display in the .DB file on disk. Each time the display is opened, it will appear the same as last time.

Label

Within the requester, a series of controls exist which allow you to edit the label text, with the column's width, format and alignment.

Label Text

Amongst other things, the text of the label may be changed. This can be a very useful feature since changing the name which appears in the List mode display does not affect the name of the field in any other way. When returning to the Record mode for example, the label will be the same as it was before editing it in the List mode. This will allow you to shorten certain headings from a verbose legend to one which will provide more columns across the width of the screen.

For example, a column called **Quantity** may have been defined as a right aligned, 3 digit field. When in List mode, the visible heading may be the last 3 characters of the name i.e. **ity**. By editing the field heading, this truncated label may be replaced by a more useful and recognisable abbreviation such as **Qty**.

Output format

For numeric headings, this control allows you to determine the style and precision with which values in this column are to be displayed. The current setting for the numerical display is shown to the right of the format button.

Alignment

Data in the columns may be aligned with the left or right hand edges or it may be centred between them. Use the cycle gadget at the bottom right to effect changes.

Appearance

The three radio buttons at the top of the requester tell Twist which information you wish to edit. It is possible to make the headings appear in a very different way from the data which is displayed beneath. Before editing the display, select the mode you wish to alter. **Label** edits only the label header section, **Record** the record lines, **Both** edits both at the same time.

Controls

A few miscellaneous controls are available.

Change all

If **Change all** is selected, when clicking OK, the newly determined format will be applied to all fields in the List mode display. With this unselected, the changes will only be applied to the field which was double-clicked on in the first place.

Delete

It is possible to selectively remove columns from the List mode allowing you to simplify the display by removing unwanted fields from a screen which is not wide enough to cater for a large number of items. You will be prompted for confirmation before the operation proceeds.

New

This will allow you to insert or re-instate fields which have been removed from the display using the **Delete** function. The new field is inserted into the column which was clicked on when entering the mode. The old field will be moved over to the right by one position and its display parameters will be left intact. The new field which is to be inserted will assume the parameters in the requester.

Upon clicking on **New**, a pop-up list of the fields will be displayed. At the foot of this display another item labelled <Change File> will be seen. Selecting this will replace the pop-up with a list of other connected databases. This function will allow you to insert fields from other files into the display.

Mark record

Invert

Mark All

The **Mark record**, **Invert** and **Mark All** commands are used in conjunction with **List mode** to mark records prior to using them in a report or other filtered operation.

Mark record marks the current entry, **Invert** inverts the selection mark and **Mark All** selects all records.

You can also mark a record in **List mode** by left-clicking at the left-hand edge of the record.

Marked records are shown by a dot at their left edge.

Import

AM

Export

AN

The **Import** and **Export** options open a window for importing and exporting data respectively; this is discussed in the section *Import/Export*.

Update

AU

The **Update** option opens a window for processing an update; this is discussed in the section *Update*.

Mailmerge

AJ

The **Mailmerge** option opens a window for processing a mailmerge; this is discussed in the section *Mailmerge*.

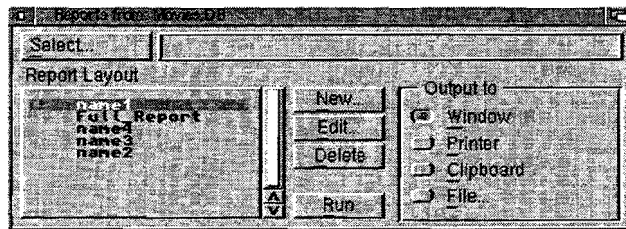
The Report Menu

Edit report...

AP

This menu opens a requester showing the list of defined reports, allowing you to create new reports or edit existing ones. The first 8 defined reports appear in the menu itself allowing you to run them directly from the list.

If you choose **Edit report**, you will see the following:

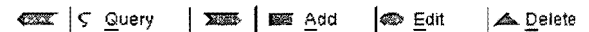


the Reports requester

From here you can choose a report, edit, delete or run it or you can create a new report. More detail on reports is given in the *Querying and Reporting* section.

The Buttons or Action Gadgets

Below the menu line Twist has a panel of buttons, or action gadgets. The buttons on the panel change depending on what you are currently doing. If you are in the main mode the following buttons will be available:



Finds the previous record. If shift-clicked, it takes you to the first record in the database. If the database is very large and contains thousands of records, then moving to the beginning may take a few seconds. This may be aborted by pressing **Ctrl-C**.



Allows you to define a query. A query is used to find records in the file. As Twist enters query mode the panel of buttons changes. See the section *Querying and Reporting*. If **Query** was already selected i.e. the window shows only records that match a previously defined query, then Twist deselects the button and shows all of the records.



Finds the next record. If shift-clicked, Twist takes the display to the last record in the database, in which case if the file contains many records (thousands) it may take a few seconds to move from one end to the other. This operation can be aborted by pressing **Control-C**.



This function allows you to add new records into the database. In response, the button panel changes to provide the following:

Within this display it becomes possible to enter data into the fields. When a record is complete, select **Save**. At this point, Twist saves the input as a new record and immediately empties the display, allowing you to create another record.

If you are about to create a record which carries some data across to the new record, you may use the **Duplicate** button instead of **Save**. **Duplicate** saves the current record but does not empty the record form.

After you have saved the last record, select **Cancel**, this returns the screen to the main data display mode.

Selecting the close box exits Twist, with a warning first.



Like **Add**, this also enters edit mode but it is used to alter existing records. Within this mode; **Save** saves your changes and returns to the main screen. **Cancel** abandons the changes. **Duplicate** saves any changes to the edited record and allows you to edit a new record with the same values. **Edit** can also be activated by pressing **Return**.

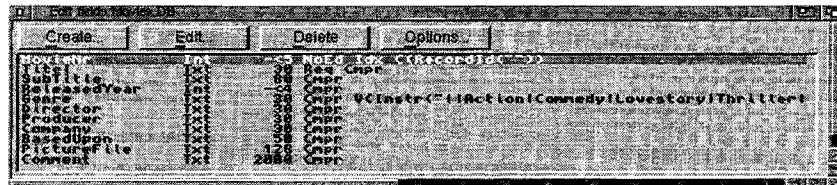


This function deletes the current record. If you are in **List** mode and have marked one or more records, then this function will delete those records instead. Since deleted records can't be recovered you are asked to confirm your intention first.

Setting up a Database

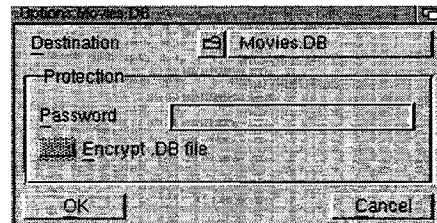
Database Fields

The fields in a database are what define the structure and content of a database; in Twist these are set up via the **Edit fields** command from the **Project** menu; this is entered automatically when you select **New**:



The action gadgets control field editing: new fields may be created via **Create**, existing ones modified via **Edit** or deleted via **Delete**.

The **Options** gadget allows the destination file for the database together with a password to be set:



If **Destination** is selected a new file may be created, so that instead of overwriting the old .DB file a new file is created. This is often recommended when performing experiments on valuable data.

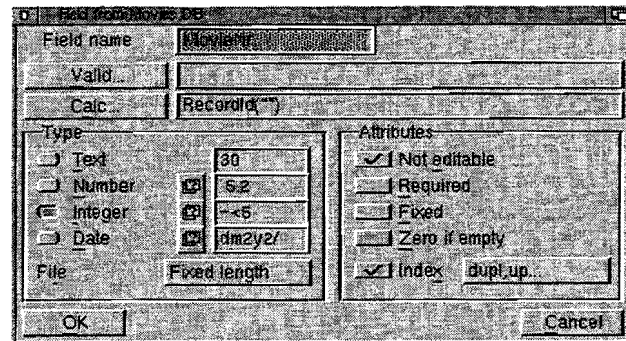
A database file can also be assigned a password; before opening the file the user will be prompted for the password. Passwords can be up to 15 characters long and may contain any characters. Be very careful not to forget the password, otherwise the data will be lost forever. The password is saved in encrypted form in the .DB file.

In addition the data within the file may be encrypted; if a password is assigned and encryption enabled then the data is saved in a scrambled format. Thus not even a hexdump of the file would show any meaningful data.

Editing fields

AF

On creating or editing a field, the following requester is used:



Firstly the field must have a name. This can be any meaningful (or meaningless!) string of characters, up to 15 characters long. A field name cannot include spaces, but Twist will accept the underline '_' instead. When entering data, it is easier to remember (or work out, if someone else created the database) what is supposed to go in each field, if the field names reflect the expected data. For example, Name is more readily identifiable than Field_one, or aabbcc, but if the name has been split into two fields, one for surname and the other for any forename(s) or initial(s) then Surname is more explicit and is less likely to have other users entering the wrong data into that field.

For a new file, simply entering the field name and pressing Return or Tab creates a text field 30 characters long, as that is the default field format. Once a field with a different format has been created, it will become the current default. This can be useful if you want to create several fields of the same type and length.

Field Types

When a field is created, Twist must be told how it is to treat the data that is to be entered: this is known as giving the *field type*.

Fields can be either free-entry or calculated and/or validated.

Free-entry fields

This means that the user enters the data required into a field and as long as it fits with the format (i.e. not entering text into a number field), the program will accept the input.

Calculated fields

With a calculated field, Twist calculates the data for that field, either from the input, or more commonly from two (or more) other fields. For example, on a database relating to customer orders you might have:

Quantity * Unit Cost = Total cost

Calculations for VAT are another common use for this type of field.

Calculations can be carried out on dates as well, for example subtracting date of birth from current date (Today ()) gives age. Divide this by 365.25 to get the age in years.

Calculated fields are set up by selecting **Calc** from the *Field definition* requester. This gives a requester with a list of operators, a list of fields and a numeric keypad, all of which can be selected by use of the mouse and will then appear in the formula box, or the formula can be typed straight in. So in the example above, you would select **Quantity** from the field list, ***** from the numeric keypad or the list of operators and **Unit_cost** from the field list, to define the calculation for the **Total_Cost** field.

Validated fields

A validated field is one where the data entered must meet some expectation, otherwise it is not accepted by Twist and an error message appears; the user must then edit the entry.

In order to create a validated field, select **Valid** from the *Field definition* requester. This gives the same calculation screen as for calculated fields. Into the formula box you need to enter the criteria for that field. For example, to validate a field for VAT, the formula would be

VAT = 17.5 || VAT = 0

Note: A field name is not accepted as part of a formula until the field has been saved, so to use the field name as part of the validation formula for that field, the field needs to be saved and edited, rather than validated when it is first set up.

Data type

There are four types of data that can be entered in a field: text, number, integer and date.

Text fields

A text field can be between 1 and 30000 characters in length. It can accept letters, digits, spaces, hyphens. Any digits entered in a text field are treated as text. This means that they will be treated as text for the purposes of sorting or searching (see *Screen Queries - Find text*).

This fact also means that it is not possible to directly carry out calculations on text fields. For this reason a number of facilities are provided allowing you to convert numerical text into its direct value equivalent for the purposes of evaluation, comparison and calculation. Of course, it is also possible to convert numerical values back into text. These conversions are handled mainly through the `Str()` and `Val()` commands. These are described in more detail in *Appendix A - Functions and Expressions*.

Note that if you are using long fields (say more than 100 characters) then setting the **File** option to **Compressed** will save space in the resulting database file, at the expense of some performance loss - the various file options are discussed later.

Number fields

A number field will hold a value of up to 13 digits in length, plus up to 8 digits after the decimal place. The default length is 5 digits and 2 decimal places. A number can be either signed or unsigned. If it is signed, then the positive sign can be a +, space or nothing. The default mode for a signed number is a space for positive value and a - for a negative. Unused characters before the decimal point can be filled with spaces, *, zeroes, or left unfilled. The default is that unused digits contain spaces.

To change the defaults, left-click on the gadget to the right of **Number**. This brings up a requester which shows the different options and a sample output. Each option can then be selected and changed.

Integer fields

An integer field is numeric, but without any decimal places. Field length can be up to 13 digits. Again there is a choice of signs and space padding, the options and defaults are the same as those for a number field. To change the defaults, left-click on the gadget to the right of Integer. This brings up a requester, with the options and a sample output.

Date fields

The date field is used for storing dates, and there are a number of formats available. The sequence can be in Day, Month, Year, or Month, Day, Year or Year, Month, Day. The delimiters between each portion of the date can be either /, or -, or . In addition the year can be shown as either 2 or 4 digits and the month can be either 2 digits, three letters or the full name.

The default is DD/MM/YY e.g. 24/12/93. To change the default, left-click on the gadget to the right of Date to bring up a requester.

The space taken by a date field is four bytes. This is because Twist stores all dates as the number of days which have elapsed since AD 0 January 1st and this value is held as an integer.

It is possible to validate and perform calculations on dates by converting them using the string handling commands. `Str()` and `Val()`.

Field Attributes

In addition to the field *types* there are four *attributes* that can be applied to any of the types. Not editable, Required, Fixed and Zero if empty.

Not Editable

Not editable means that the data in that field cannot be edited by the user and can therefore be used with a calculated field, for example. When entering data, the cursor will skip over any Not editable field, so it is impossible to actually enter any data into it.

Required

If a field is Required it means that the record is not valid unless some data has been entered into that field. Trying to save a record without entering required data will result in an error message. The user will then need to return to the data entry screen and correct the record.

Should you wish to create a required field after some data has been entered into a database, a little thought shows that this causes a problem. You create a required field but none of the records will have data entered into that field yet! There are two ways round this.

You could first create your field without the **Required** attribute, enter all the data and then change the field to be required. Or you can create the field as not required, issue an update to change all occurrences of this to a particular value (see the **Update** section) and then change the field to be required. This latter method is obviously much faster if a great deal of data has already been entered.

Fixed

If a field is fixed then, after the first time data is entered into the field, no further calculation or validation will be performed.

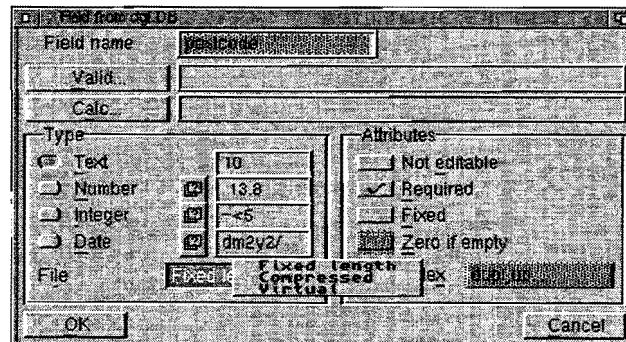
One application for this is for entering a date into an order. When the order is initially created, then a date may be inserted into the field, subsequent accesses or additions to the order will not update this field.

Zero if empty

If a numeric field is left empty, normally the result of working with it will be undefined (as defined by the **IsUndef** function); but if the **Zero if empty** attribute is set then Twist will force the field to be zero if there is no valid data.

File

The pop-up allows the manner in which the field is stored within the database to be set:



The normal setting, **Fixed length**, stores all parts of every record entered in the database file whether all characters are used or not; obviously if you have large text fields the size of the database can rapidly become extreme.

The **File pop-up** provides a method of avoiding this, with the **Compressed** field. If a field is set to compressed then, rather than being stored in the main database file, it is stored in a second file which holds all of the compressed records in a database together, thus saving space. This has the disadvantage that the time required to load a particular field is longer than for the fixed format (since Twist cannot immediately find it in the main file).

The third setting, **Virtual**, stores nothing at all in the database file; instead the content of the field is computed every time the record is used from the calculation attached to the field. Note that a virtual field without an attached calculation doesn't make much sense!

Index

This check box brings us to the wider subject of indexing a field which will now be described in more detail. A field may be assigned the index attribute either within the *Edit fields* requester or by choosing **Index...** from the **Project** menu.

Indexing - Overview

A field can be indexed. This means that a separate file is created which contains a copy of all the values for a field sorted into either ascending or descending order. For this reason, indexing fields can produce dramatic speed increases when processing queries or generating reports. The index files have the same filename as the .DB file, the only difference being that the file extension is a field number. For example, **address.000** would be the index file for the first field in **address.DB**.

The size of the index file is dependent on the field length and the number of records. It is recommended that you don't create indexes for very long text fields, since they will be slow and huge and this is contrary to the reason for creating the index in the first place.

If many index files exist for a database, then the total size of these files may easily exceed that of the .DB file itself. For this reason it is sensible to be selective about the fields which you wish to index, don't index fields that you don't really need. In case you're worried, an index can be deleted or created at any time and does not contain any data which is not already in the main database file.

When Twist uses an Index

Twist uses field indexes for 3 purposes.

Display sequence

The active index defines the display sequence used when scrolling through the records. If there is no specified index, then the records are displayed in the order in which they exist inside the computer. This is usually the order in which the records were originally created. For large files this may be quicker for the List mode, but Twist is so fast that speed is rarely a problem.

Queries

When searching, Twist examines the query string. If this contains a reference to an indexed field (and other conditions are satisfied) Twist will use that index field to speed up the search. In many cases this means that Twist can search through thousands of records in fractions of a second.

Relations

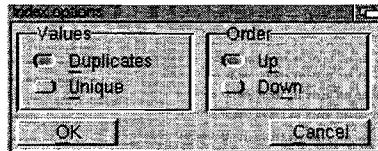
When performing reports, updates or mailmerges, it is possible to link fields from different database files together. Such links are known as *relations*. Relations usually exist when one database (the source) contains data which another (the destination) wishes to use.

For example, this might be useful when setting up a database for stock control and sales. As a sale from one database reduces stock as it is sold, then another database may store the re-ordering information to generate a suitable order for the vendor. The common information which may be shared will be the product order code or stock description and the quantity.

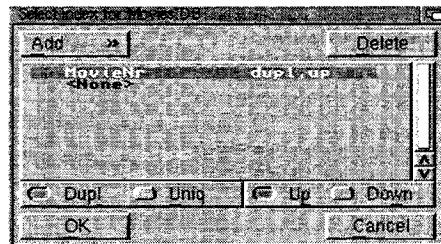
To create a relation between two different databases, it is necessary that at least one field is shared between the two (i.e. same field name and data type etc.). This field should be indexed within the database which wishes to use the data (the destination) so that Twist can perform a quick lookup.

Indexing a field

To make a field indexed you can either check **Index** within the *Edit field* requester or you can select **Index...** from the **Project** menu and add a field. In either case you will be able to choose various *Index options*.



Index options from within the Edit field requester



Index options from Index... on the Project menu

Field index status

When created, each index field is assigned two attributes. These govern the way in which Twist treats the data contained within the resulting index file. These attributes can be changed by clicking the appropriate radio buttons for the appropriate index.

Dupl

Short for duplicate, this function will allow duplicate field values.

Uniq

This button forces Twist to check each time that a record is added or changed that no other record has the same field value. If this is not the case you are prevented from saving it.

Up or Down

The contents of a field index file are sorted into ascending (increasing) or descending (decreasing) value order, depending upon the status of these buttons. Up is ascending mode and Down is descending!

If you choose Index... from the Project menu, you can define and delete indexes as follows.

Using Index from the Project menu

Add

To add a field to the index list, simply click on Add and select the required field from the resulting pop-up.

Once a field has been selected, it will appear in the centre of the display along with any other index fields. Initially the newly indexed field will be assigned two control attributes, these are visible to the right of the field name as Dup1 and Up. These controls are discussed in more detail in the section, *Field index status*, above.

Delete

Click on this button to remove the selected index from the list.

Clicking OK

New indexes in the list are indicated by an N to the left of the field name. When you select OK, the index files for these are created. If other indexes have been modified or removed, their files are rebuilt or deleted from the disk drive at the same time.

Editing the Record Form

Edit form...



The record form defines how records appear in record mode and is used to both edit a record and to define a query. Choose Edit Form... from the Project menu.

Nr	FirstName	Surname	Sex	Fee
Actor 1	Actor 1 First Name	Actor 1 Surname	Actor 1 Sex	Actor 1 Fee
Actor 2	Actor 2 First Name	Actor 2 Surname	Actor 2 Sex	Actor 2 Fee
Actor 3	Actor 3 First Name	Actor 3 Surname	Actor 3 Sex	Actor 3 Fee
Actor 4	Actor 4 First Name	Actor 4 Surname	Actor 4 Sex	Actor 4 Fee

Edit Record Form

Although the record form is saved as part of the database file, it can be changed independently of the field layout. When a database is first created, a default form is created at the same time, normally this is then modified by the user to create a custom form; later changes made to the field layout do not affect the custom form (by default).

Form Design

Objects located on the form may be selected, moved or sized in the work area. Before anything can be done to an object, it must first be activated; this may be achieved by clicking on it once with the mouse. An active object is indicated by a square in the bottom right hand corner of the box.

This square is used to alter the size of the object, hence from this point forth it will be referred to as the *sizer*.

If you want to manipulate multiple objects they can be shift-selected (selected whilst one of the **Shift** keys is held down); if an object was previously un-selected it will be added to the selected objects, if it was previously selected it will be removed from those selected.

Twist also allows 'rubber-banding' around a number of objects; point at some clear area of the form, hold down the mouse button and drag - all of those objects which the rubber band touches will be selected.

Resizing

To alter the size of an active object, simply move the mouse pointer into the bottom right hand corner (over the square) and press and hold the left mouse button. While the button remains pressed, the mouse can be moved around the screen and an outline of the object will be seen to follow. When the new size appears satisfactory, let go of the mouse button. The object will be redrawn with its new border size.

Moving

To move an active object, simply place the mouse pointer somewhere over the centre of the object and press and hold the left mouse button. Moving the mouse will now cause an outline of the object to move with it. To release the object at any position on the screen, simply let go of the mouse.

Appearance

In a record form, there are five types of object which can be used, these are Field, Text, Box, Image and File. The appearance of any object may be edited by double-clicking on it. Depending on the type of object chosen, a requester will appear on the screen allowing certain attributes for the object to be edited.

Adding new objects

To add new objects to the record form, simply drag (left-click and hold) the appropriate symbol from the left of the work area and drop it in the appropriate position within the window. Text, box, file and image objects are dragged directly from their icons on the left and released, whereas dragging the field icon will pop up a list of fields not already contained in the form when you release the mouse button; choose a field from the list.

Of course the default form already contains all the fields so, if you have just started to edit the form, there are no fields to add. When placed onto the form, field objects have their field name displayed within them and the surrounding brackets indicate the character length of that field.

Deleting objects

Objects can be removed from the form by dragging them to the Trash symbol in the window (not on the Twist desktop). If you drag a field object to the trash can, the corresponding label will also be deleted.

Boxes, Backgrounds and Groups

If you add a box to a form, you will almost certainly need to enlarge it afterwards by dragging its sizer symbol. If objects are dragged onto a box they will be grouped together with that object.

Boxes may be used for two main reasons. First of all, because they can be filled, they can be used as a decorative background for data entry forms. The selection of background colour and fill pattern may be chosen to complement the colour of labels used in the presentation.

Alternatively, because of the way in which Twist groups objects which are placed together in a box, record forms can be easier to manipulate when only the box needs to be moved, since moving the box will cause all objects grouped with it to be moved at the same time. Incidentally, this also applies to deleting the box, so beware!

Overlapping objects

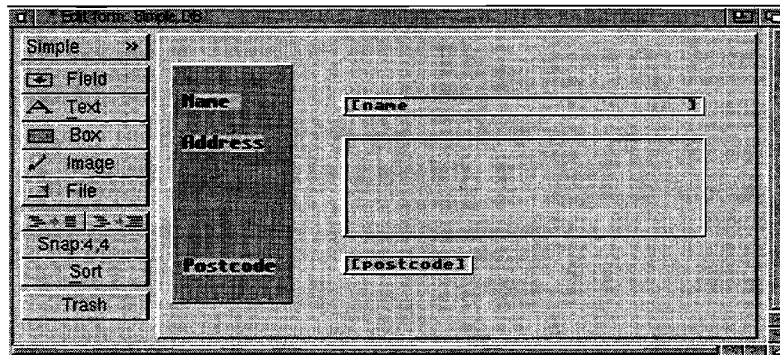
After placing a box on the screen and enlarging it, the chances are that you will have covered other objects.

Long-clicking

It is possible to push the box to the back of the display by clicking on it once and holding the mouse down for slightly longer than you would normally, say for half a second - the mouse pointer will change to a clock icon. This is called *long-clicking*. The box will then be *sent to the back*; you can bring it forward again with the same operation.

You can perform this ordering of overlapping objects by long-clicking on *any* object; to get it right every time takes a little practice but it does let you arrange the form in almost any way you like.

There is something special about box objects: when you send a box to the back then any objects that are totally enclosed within the box become its *children* and will be moved with the box when it is moved. If you bring a box to the front it will lose its children and all the objects can then be moved independently again. Here is a box with 3 children:



Objects contained within boxes are not *true* children of the box but it is a convenient word to use. If you delete a box that contains object children, all the children will also be deleted from the form.

Sorting objects

Objects are ordered. Two types of order are important to us: the input sequence when records are edited or entered and the physical appearance of objects which overlap each other. In Twist, these are combined into one and are not separately changeable. We have described how to send objects to the back and bring them to the front in the previous paragraph - the order so defined affects both the physical appearance of and, where relevant, the entry order for the objects.

Let's say we have a very simple form that look like this:

When entering data into fields within this record, you move from field to field by using the Tab, Shift-Tab or the cursor keys. To ensure that, say, the postcode is entered first you must send the other two entry fields (name and address, the middle field) to the back; to attain an order entry of postcode...name...address, first select **Sort** (to ensure a left-to-right, top-to-bottom order, see below) and then long-click on name followed by a long-click on address. This sends the name to the back first and then the address to the back.

To attain an entry order of postcode...address...name, select **Sort**, long-click address and then long-click name. With practice this becomes easy.

In addition to defining the input order, long-clicking can be used to place objects in front or behind each other.

Here the word **Addresses** (in white) has been overlaid at the top of the box. Long-clicking on it would send it behind the box.

The Sort Gadget

As we have said, you can sort objects manually using the long-clicking technique - this can, of course, lead to a confused display if the form is large. To restore order you can use the **Sort** gadget which sorts the entire form on a left-to-right, top-to-bottom basis.

1	2
3	4
5	

This means that 5 will be *physically* in front of 4 which will be in front of 3 etc. and the *input order* within fields is 1, 2, 3, 4, 5.

Objects within boxes will also be sorted on this basis - thus they are not true *children*, as we have said.

The Snap Gadget

As we have seen, it is possible to move and resize active objects by clicking on them and dragging them with the mouse. You may have noticed that, in so doing, the objects tend to jump around the screen in small but noticeable increments, these jumps are known as a *snap*.

You can set the snap distance by clicking on the **Snap** gadget and entering both a horizontal and vertical snap distance, which may be different. Ensure that **Use raster snapping** is checked.

Using the Keyboard

As well as using the mouse, active objects can be moved and resized using the cursor keys on the computer keyboard. To move an object, simply use the cursor keys; with each press, the object or group of objects will move in the desired direction. Note that snap is ignored when using the cursor keys, this allows for precise positioning of objects.

Resizing objects is just as easy. Simply hold a **Shift** key down while using the cursor keys. This will cause the object(s) to expand or contract in the desired direction. Again, snap is ignored.

Using Objects on the Form

We have covered the general aspects of form design - now we are going to describe the individual objects that may be placed on a form.

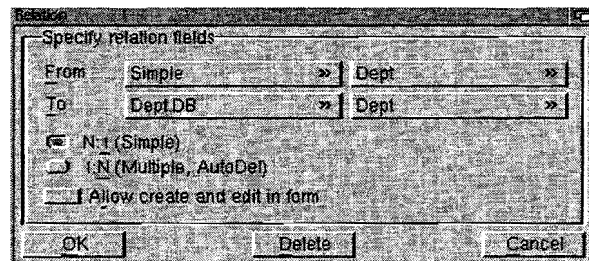
Field objects

Field objects are used to enter data into the fields of a database, normally the default form will include all of the fields used by your database so they can simply be edited by double-clicking on them.

Field objects are added by selecting the **Field** gadget from the form editor. Hold and drag this icon on to your form in the desired position and let go of the mouse button; a pop-up of field names not used so far will appear. Select the required field name.

Flat-file database forms are built up using just the fields contained within the database but what if you want to build relations into your form?

At the top left of the gadget bar you should notice a gadget that has the name of the file for which you are creating a form - click on this and you will see a pop-up of the names of any connected files and <New File>. Choosing this allows fields from another file to be selected and placed on the form. Selecting a file will bring up this requester:

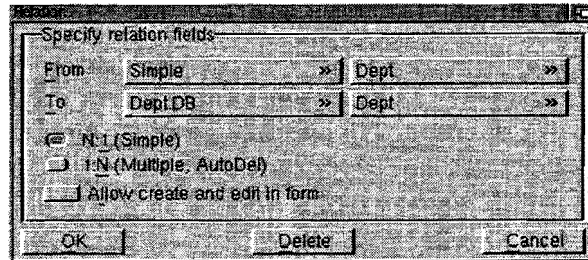


The **From** line shows the file and field from the main file which is to be linked to the related file, shown on the **To** line with its file name and field. You can choose the filename and field name for both **To** and **From**; **Twist** will try to be intelligent in guessing which field you want in the **To** database, knowing the requested **From** field.

The relationship between the databases can be N:1 or 1:N (you can also build up an N:M relationship using an intermediate database).

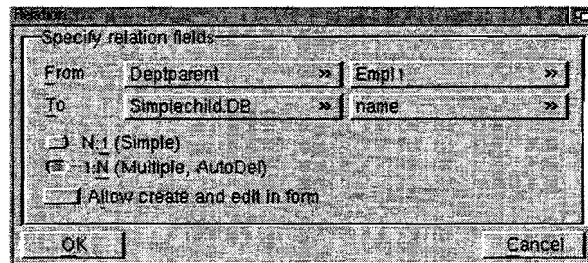
N:1 Relations

N:1 relations are used to link multiple From records to a single To record (for example many employees to a single department):



1:N Relations

1:N relations are used to link multiple records on the form to records in another database (for example to link a department to several employees):



For this kind of relational join Twist will automatically delete dependent records in the related file if the parent record is deleted, for example if a department is closed then all the employees for the that department would be orphaned and therefore deleted. It would be kinder to allocate them to another department before removing their old department!

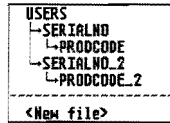
Whether a relation is 1:N or N:1 often depends on how you are viewing and organising your data.

Allow create and edit in form

This option allows new entries in the child database to be dynamically created whilst entering data into the parent database; for example in a database of users and serial numbers, creation of serial numbers (in a separate database) is much easier if they may be entered while entering the user details into the user database.

Using the relational fields

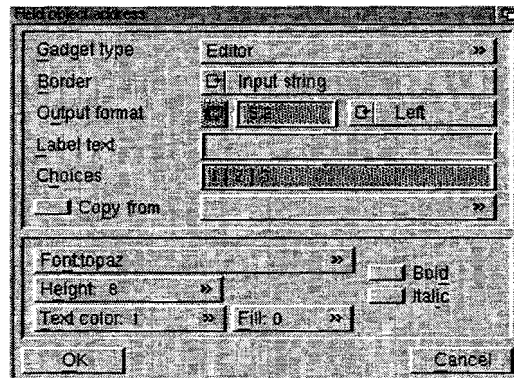
After setting up the various relations, complex interactions (1:N->N:1) are possible, with multiple entries, for example structures such as this are possible:



Any of the fields from these files may be used on the form, or the file interactions modified by selecting the appropriate file. Note that if there are multiple links to a single file via an N:1 relationship Twist automatically numbers the files _2, _3 etc.

Field objects

The various field objects used by Twist may all be edited by double-clicking on them; this brings up the *Field object editor* requester:



The requester contains 2 sections and allows access to the options for the five varieties of form object:

The top section of this requester allows you to choose the type of gadget, the output format etc., for the field.

The bottom section of the requester allows you to choose the font for the input string (and any directly associated label text) from the installed system fonts. You can change the font name, its size, colour and text attributes.

Border

You can choose the border around input fields from a large number of different types of border. If you want to design your database according to official Amiga guidelines, you should consult the *Amiga User Interface Style Guide* before choosing a border.

Output format

Depending on the type of object you can choose the numeric/date format and the alignment of the object (left, right, centre).

Label text

The default label of a text object is its field name, this saves you the trouble of having to create a label for every input field.

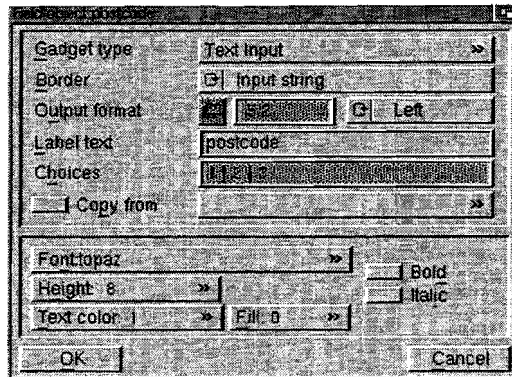
Copy from

This pop-up may be used to set a default value for a field before the user starts entry on it; this might be used to set the standard discount rate given to a customer, whilst allowing it to be modified for a different value. It may also be used to copy from a related file (say the name of an employee's boss, given his department).

Gadget type

The type of gadget defines the way in which data is entered into the database. Choose your required type from a pop-up; the options are Text input, Pop-up Menu, Radio Buttons, Check Box and Editor.

Text input



This is the default method of entry used by Twist. Normally an element of data would be given two fields, a label (the label text) and the data itself (known as the data field). When creating the fields in the first place, Twist will always select this as the standard mode of entry and display, with a text and data box for each field.

Editor

The Editor object allows free-format text to be entered into a field; the classic use for this is an address.

Pop-up Menu

This allows a pop-up list of selections to be displayed as part of the data entry field; the input field will have a small >> icon to the right that shows a pop-up list when it is selected while editing records. Since the input field remains editable other values not occurring in the list may be entered by the keyboard.

Radio buttons

The Radio buttons object allows a short list of selections to be displayed as a group of radio buttons. When showing records in the database window Twist selects the option that corresponds to the field value. If the record contains a value not occurring among the options no radio button will be selected. When editing records it is not possible to enter other values than those being displayed.

Check box

The Check box object allows a field to be displayed as a simple on/off check box. A check box is used for fields whose value is either 0 or 1. If the value is 0 the check box is shown in its non-selected state otherwise it is shown selected. The label may be edited directly since for a check box it is part of field object itself and not a stand-alone object.

Text objects

Text objects are used to display normal text on a form; in addition the labels attached to fields are text objects.

To add a text object drag a text object onto the form, to edit an existing text object, double-click on it:

The Label text box is used to set the text which is to be displayed, the default of which will be the field name of any associated field.

The font, size, colour and text effects can be set in the same way as when editing a field object.

The Transparent check box allows a text object to be made transparent i.e., any underlying objects are allowed to show through.

Box objects

Box objects are used for boxes on screen; this may be to provide visual grouping of other objects, or functional grouping (since Twist treats objects inside a box as part of the box).

A box object has a fill colour and a border, according to your choice. If both the fill and the border are turned off, then the box will be invisible in the record form.

You can also associate an ARexx script with a box; this effectively gives you an ARexx button. Choose an ARexx script to assign to the box by clicking on the file selector gadget.

Image objects

By dragging the Image gadget onto your form and double-clicking on the resulting gadget, you can display icons on your form. For a more general image support you should use a File object.

When you double-click on the Image gadget on your form, a file selector will appear - choose the required icon (with a default extension of .info); if you load a large icon you may need to re-size the box to see the whole icon.

File objects

File objects allow external files to be viewed and used as part of the form.

In general, Twist uses DataTypes for its file object support. The DataTypes library is available with Workbench 3™ and above and we would highly recommend upgrading to this version of Workbench, especially if you want to use sound samples, anims and other datatypes from within Twist.

Twist *does* support the IFF picture format directly so that you load IFF files onto your form even if you have an earlier version of Workbench. If your machine supports DataTypes then Twist will use this library by default to display IFFs; you can turn this off by deleting the USE DATATYPE line in the Twist.inf file - Twist will then use its own IFF display code which is substantially faster.

The file may either be viewed as part of the record window, by selecting the .DB window radio button, or in a separate window, by selecting the Another window option. In the latter case and under Workbench 3 the file will be passed to MultiView for handling whereas under earlier versions of the operating system Display will be used, which only handles IFF pictures.

Alternatively you can specify the application to which the file will be passed for handling by editing the Twist.inf file (under the heading ; this allows tremendous flexibility i.e. you may want to use

The file displayed may be selected either as **Filename by field** - a file is selected based on the value of a text field or **By file selector** - a single file is displayed in the form based on a static file name:

This is a very powerful and flexible part of Twist; please take some time to learn how to use external files.

Querying and Reporting

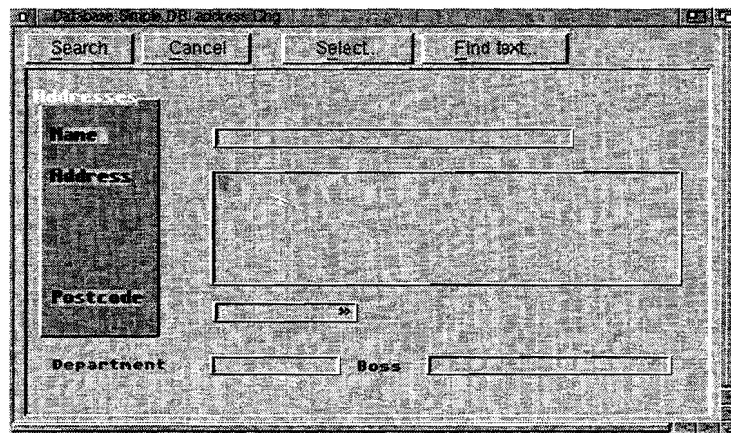
Once you have set up your database and entered some data, you will want to start getting useful information from it.

There are a number of ways of doing this, from running a simple screen query, to printing a tabular report or one that is formatted, with text incorporated into it. These *reports* can be directed to the screen or the printer, or to another file.

Screen queries output only to the screen and the output is in the form of complete records, with a choice of Record mode or List mode for the display. Reports display only selected fields, which can be arranged to suit the requirements. There are also more advanced options available on reports, such as relating two files, using the mailmerge facility and building in calculations.

Screen Query

This is the simplest way to quickly obtain specific information. From the record screen, select Query. This gives you a record selection screen as shown here:



The screenshot shows a window titled "Simple DB address Dmg". At the top, there are four buttons: "Search", "Cancel", "Select", and "Find text.". Below these buttons, the window is divided into a left sidebar and a main content area. The sidebar, labeled "Addresses:", contains four labels: "Name", "Address", "Postcode", and "Department". The main content area contains four input fields corresponding to these labels. The "Name" field is a single-line text box. The "Address" field is a large multi-line text box. The "Postcode" field is a single-line text box with a small "X" button to its right. The "Department" field is a single-line text box, and next to it is a label "Boss" followed by another single-line text box. The window has a standard Windows-style border with a title bar and a scroll bar on the right.

There are four options, Search, Cancel, Select... and Find text...:

Upon entering query mode, the input cursor is located in the first input field. If a query has been previously defined, then the entered values are displayed and the cursor is placed in the first defined field. While in this mode, the input fields will behave differently from when editing records. In addition, fields that are defined as **Not Editable** or as **Calculated** may be edited. The input length of fields is no longer limited by the field definition.

To perform a query, enter the value of interest into the field to be searched. In text fields you may also place an asterisk, *, at the start or end of the value. The * is known as a *wild* character and is used by Twist to represent a number of arbitrary characters which precede or follow the text of interest. See the description of the Like operator for further information.

Using radio button object fields

If a form has fields displayed as a group of radio buttons, then these can be used to define the search criteria by simply clicking on the radio button for which you wish to search. By default, none of the radio buttons are selected meaning that no criteria for that field is currently defined.

Database Sample DB - name Chg

Search Cancel Select... Find text...

Name

Address

Postcode 92

Department

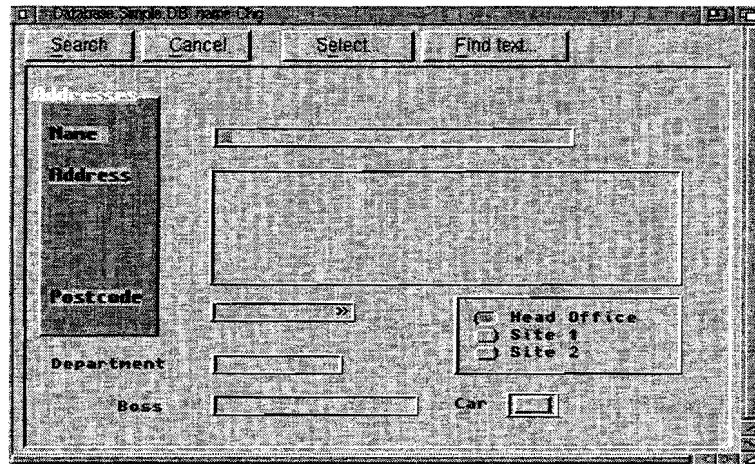
Boss

☒ Head Office
☐ Site 1
☐ Site 2

If you had selected a radio button the last time you defined the query string, then that radio button will remain selected. For input objects, the location of the input cursor will help you to remember the last field used for searching but this is not the case for radio buttons since the input cursor can't be located in a radio button object. Therefore it is important not to forget to turn radio buttons off before performing another search. This may be achieved by holding down a shift key while clicking on the radio buttons which are no longer required.

Using check box object fields

When entering the query mode, the default state for a check box is grey. This means that no criteria for that field are currently defined.



With subsequent clicks on a check box, the contents will cycle through three different states. These are ticked (selected), empty (unselected) and grey (undefined).

Building a query string

Before Twist performs a search, it builds an internal version of the query string. This is necessary because when defining the string yourself you may have taken certain liberties with the way in which the string is written. When working on a query however, Twist must examine this (possibly abbreviated) query and convert it into the full expression with the following syntax:

`<fieldname> <operator> <value>`

For text fields the default operator is the Like operator (#). The value may be entered without being enclosed in quotation marks (" "). For field types other than text, the Equal operator (==) is used.

It is possible to place query strings into a number of fields. If this is the case then, when building its internal copy of the query string, Twist pulls all the strings together and joins them with the logical operator and (&&). For a record to pass the query, it is then necessary for the first query criteria to be met *and* the second and so forth.

Other operators

The default operator can be overridden simply by entering a new operator in front of the value. This will force Twist to substitute the specified operator instead of the default. The operators which may be used are:

#	Like
.	Likelyness search
==	Equal
!=	Not equal
<	Less than
<=	Less than or equal
>	Greater than
>=	Greater than or equal

Value ranges

If a search for a range of values is required this may be accomplished using Twist's range notation; this uses the [,] and ; characters to define the lower and upper values within the range.

The brackets enclose the range definition while the semi-colon is used to separate the upper and lower boundaries. So that [from ; to] is transformed to:

```
<fieldname> >= <from> && <fieldname> <= <to>
```

For example:

```
Customer_Number [100 ; 200]
```

This will return records whose customer numbers are from 100 to 200. It is possible to modify the way in which the range works by changing the direction of the bracket at the start or end of the range.

For example:

```
Customer_Number ]100 ; 200]
```

or

```
Customer_Number [100 ; 200[
```

If the first bracket is inverted it modifies its meaning to be > instead of >=. If the second bracket is inverted it is taken to mean < instead of <=. Thus the first example above will return customers 101 to 200 and the second will return customers 100 to 199.

The Query buttons

Search

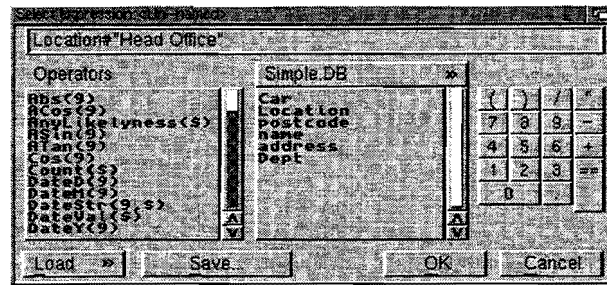
This uses the criteria you have defined to search for records and then returns to its main display mode. If the database contains many records and it is not possible for Twist to use index files to speed up the search, then it may take a few seconds before all of the records have been found. If the process is aborted by pressing Ctrl-C, Twist will display the records found so far. To the right of the database name in the title bar it shows you how many records were found. You can scroll through the records in the usual way since Twist has built a table of matching records internally, for this reason scrolling through the list is very fast.

Cancel

Returns to the main mode without performing a search.

Select

If you need to define a more complex query string than that which can be produced in the input fields, then **Select** opens the *Select expression* requester.



This requester is also used when running reports, updates etc. If the record form contains search criteria, the corresponding query string is shown in the formula box. edit the query string as required and click OK to start the search. For further information, see *Appendix A*.

Clicking on a field or operator from either of the lists inserts the selected text into the query string at the input cursor. If you click on a function, then you will also need to replace the argument indicators (9 for numerical arguments and \$ for text) with the real arguments. Of course, expressions and field names may be entered using the keyboard but the mouse may be much faster and is less prone to error since it eliminates many opportunities for typing errors. The two lists may also be scrolled using the usual cursor movement keys.

When OK is selected, the query string is syntax checked. If there is a problem Twist displays an error message and locates the input cursor near the cause of the problem.

Saving and loading query strings

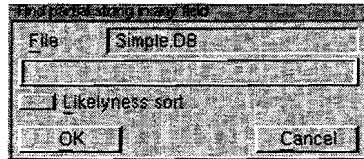
Once the formula has been entered it is also possible to save it. To do this select Save instead of OK from the formula definition screen. You will be prompted to enter a name. The name can contain up to 40 characters; spaces, hyphens and underlines are all allowed.

Once the expression has been saved, it can be selected again, by selecting Query then Select. This then gives you a pop-up list of saved expressions, or the option to Edit expression. Selecting a named expression will automatically list records in that selection. Selecting Edit expression will allow either a new expression to be set up, or an existing expression to be edited.

To edit an existing expression, select LOAD from the formula definition screen. This will bring up a list of all saved expressions, you can then select the one you want to edit.

Find text

Selecting Find text gives you the opportunity to do a free text search; that is, the program searches all text fields in all records and lists for all those records in which the word or words appear:



It is also possible to search on part of a word, so listing all records with that word in whatever form, e.g. relat, would list records containing, relate, relation, relative and so on. It is not possible to search for numbers, but digits included in text fields are searched.

The selected records are then listed on the screen in either record mode, or list mode, depending on which has been selected. When you have finished looking at this selection of records, clicking on Query again restores the list of all records in the file.

The Likelinessort check box can be used when you wish to look something up where you are uncertain of the spelling, or where you suspect the record may be mis-spelt. It attempts to search for likely records placing those it considers most likely at the top of the list.

About Reports

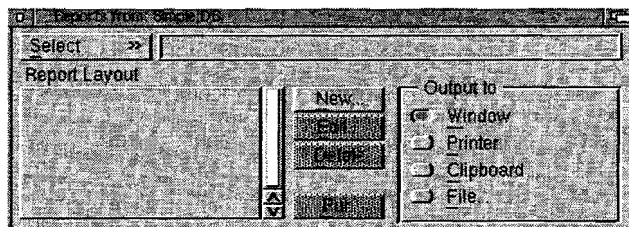
Reports are used to extract information from the database. If you want to print information from the database a report is the means by which to do so. Reports can be sent to the printer, opened as a temporary document by Twist's integrated text viewer, sent to the Clipboard or saved as an ASCII text file. The records to use for the report can be defined by a query string. Records may also be sorted and grouped.

The fields to use and their positions may be set individually for different reports and are independent of the record form or the list mode layout. Calculations and statistics may also be performed and displayed within a report display.

Setting up a Report

From the record screen, select the Report menu. The menu will list Edit report... and the names of any reports that have been set up. Initially of course there won't be any. Select Edit report... to set up a report.

This gives you the report requester, as shown here:



As you can see, the output can be directed to either Window, Printer, Clipboard or File. The default choice is Window and even if you want to print the report eventually it is worth while spending time viewing it on the screen in the first place, to check that it is going to look the way you expect.

Report Destinations

Window

This option opens the output file into a window controlled by Twist's integrated text viewer. The viewer allows you to look at the result before printing it. This will save you paper when you are creating new report layouts.

Printer

The created file is sent to Twist's printer manager which then passes the documents on to the Amiga's operating system for printing.

Clipboard

This is used for communication with other programs that support it.

File...

The result of the report may also be saved in a named ASCII file. When clicking on the radio button the file selector appears allowing you to select a filename.

New

Use this to set up a fresh report; you can then choose a report style from the pop-up. **Empty Report** gives you a blank form to work on, you design everything yourself.

Like record form defines a skeleton report that mirrors the design of your database's form.

Like list display gives you an outline report that looks like the List display, surprisingly!

Finally, you can start your report design by choosing the same layout as that of an existing report - choose **Like report...** and select a report from the pop-up that appears.

A file selector will appear and you should name your new report; Twist will suggest a default, sensible name.

Edit..., Delete

These allow you to edit an existing report or delete a report, respectively. Choose a report from the left-hand list and click **Edit...** or **Delete**.

Selecting records for the report

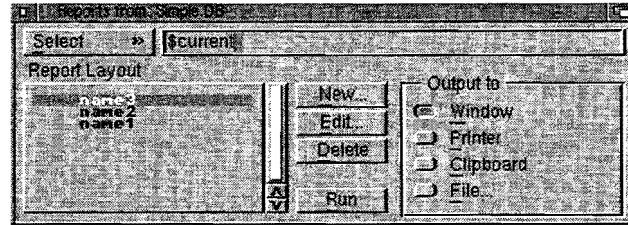
To set which records are used in a report you need to enter the selection criteria in the **Select** box. Click on **Select**, this will list any expressions that have already been set up and saved. (See *Screen Query - Saving query strings* above).

If you want to set up a new one select **Edit expression**. If there are no expressions saved, then you will go straight to the formula definition screen, with a list of operators, a list of field names and the numeric keypad. You can then enter the formula for the selection. This can then be saved, as with a screen query.

When you have entered the formula, click on OK, this will take you back to the report requester.

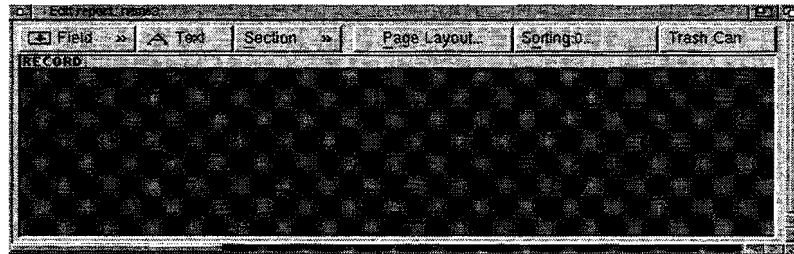
Accessing current or marked records

Sometimes you wish to print a report for only the current record (the current record is the one being shown in record mode or the record shown in List mode with a dotted frame around it). For that purpose a special query string exists: `$current` You must enter `$current` directly in the input box. Similarly it is possible to select only the marked records in List mode using the symbol: `$marked`.



Designing a Report

Having chosen to create a new report or edit an existing one, you will see the following screen.



Adding Fields

Towards the top of the screen, there is a thin white line, with the word **RECORD** in a small box. This line represents the amount of space devoted to each record that will appear in the report. To increase the amount of space, click on the word **RECORD** and drag downwards - the white area will become larger.

At the top-left of the screen is the word **Field**. Clicking on that will display a field list. Move the block cursor to the first field that you want to include, either by using the mouse or the up/down cursor control keys. Then click and hold to select that field and move the mouse on to the report area and let go. A box will appear and this can now be moved to the required position by clicking and dragging with the mouse.

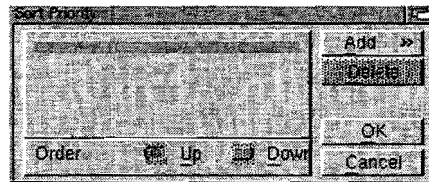
Repeat with any other fields that you want to include. This means that the fields can be arranged in a row, a column, or any other arrangement to suit your requirements.

This gives you your basic report layout; which fields are included and how they are arranged. There are other optional elements to a report. These include a sort order for the output, headers and footers, text, calculations and data from other files.

Sorting Order

In order to make sense of the output on a large report, possibly listing hundreds of records, it is helpful to be able to sort them into some kind of order, before displaying on the screen or sending to the printer.

To carry this out select **Sort Field** from the top of the report definition screen. This gives you the *Sort Priority* requester.



To add a new sort field, select **Add**. This brings up the field list and you can select the field that you want to sort on. There can be up to five sort fields in operation at any time. The top one in the list has the highest priority. This means that sorting is carried out in order of that field first. Where two or more records have the same value in that field, then they will be sorted according to the second field. Where two or more records have the same value in both the first and second fields, then they will be sorted according to the third sort field and so on.

To delete a field from the sort order, move the cursor onto it and select **Delete**. To change the sort order from ascending to descending, make sure that the cursor is on the field name and select **Down**.

If sorting fields are shown in grey this is because you have added a **Before Group** or **After Group** section. To group records they also need to be sorted in the field on which the grouping is performed. The grey grouping fields can't be deleted but they can be replaced by adding an explicit sort order for that field.

Headers and Footers

In order to add headers and footers to the report, it is necessary to add extra sections. Select **Section** from the top of the report definition screen. This brings up a list of optional sections to the report, including **Header**, **Before Report**, **Before 1st Group** and so on. Select the part of the report you want to add in and a white line, similar to **RECORD** will appear in the appropriate place.

The height of a section in the window defines how many lines are printed for that section. If for instance you wish to put each field on its own line, then you will need to increase the height of the section. To do so, click inside the dotted **RECORD** frame and drag it downwards. As you drag it, the height in lines is displayed at the top of the screen.

Before Report appears once in the output. It will follow any **Header** that has been added and come before the listing of records. It can be used for such items as the date the report was run, the report title and so on. **Header** will appear at the top of each page in the report and can be used for titles, column headings and any other details that should be repeated for each page.

Before 1st Group is a section header for each group of records according to how they are sorted. Select this and then the appropriate field name from the field list. This can contain information that you only need once for each group. For example, sorting output by customer, you may want to place fields containing information such as customer name and address in this section header rather than have them repeated on each record, which would waste report space.

Before 2nd Group works in the same way as **Before 1st Group**, but obviously for a second sort order.

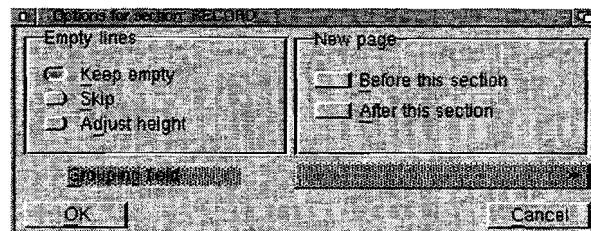
After Report, 1st Group, 2nd Group and Footer are the opposite of the Before Report, 1st Group, 2nd Group and Header sections. All these sections are optional and a report can carry none, some, or all of them. Within the grouped section the records are sorted into order, the exact ordering of which may be determined using the sort priority requester (discussed below). Note that, before printing a section, Twist ensures that the whole section can fit on the page; otherwise it is printed on the next page.

The following shows an example of the sequence of sections for a file containing records of countries and cities grouped by country (the file contains 3 records):

Before Report	{England, Liverpool}
Before group Country	{England, Liverpool}
Record	{England, Liverpool}
Record	{England, London}
After group Country	{England, London}
Before group Country	{Germany, Berlin}
Record	{Germany, Berlin}
After group Country	{Germany, Berlin}
After report	{Germany, Berlin}

Section Options

A number of options may be set individually for each section in a report; double-click on the section's "title" in the report layout (the dotted box with the title of the section) to open the requester:

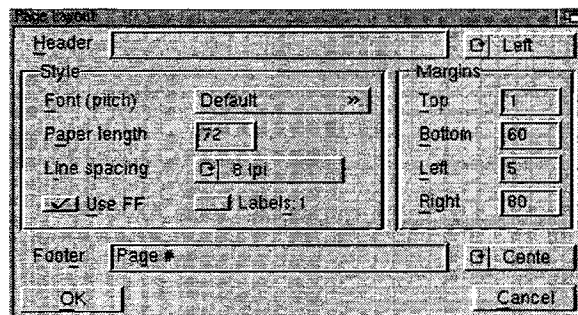


The Empty lines group of buttons define how lines should be treated that, after the record and calculations values have been added, are still empty:

- Keep empty** the empty line is printed as is.
- Skip** the empty line is not printed but the end of the section is padded with empty lines to keep the section's height unchanged.
- Adjust Height** the empty line is not printed and the end of the section is not padded, thus the section may differ in height depending on the record values.

The **New page** section, **Before this section** and **After this section**, controls where a new page should be forced.

Page Layout



In addition to the report headers and footers that can be added by **Section**, there are also header and footer lines in the page layout. To alter the default page layout, select it from the top of the report definition screen. You can now type in a header and a footer, which can include the page number. In addition; by clicking on the cycle gadgets adjacent to the words **Header** and **Footer**, they can be left-justified, centred, or right-justified. There are also options for changing the number of lines per page, the line spacing, the font and the placing of the margins.

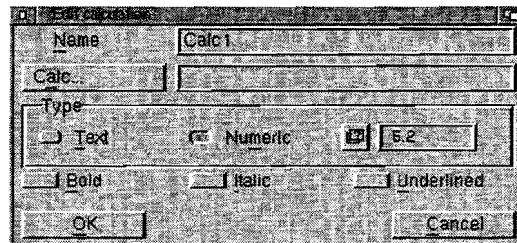
Adding Text

In order to add text to a report, select **Text** from the top of the report definition screen. Drag the resulting text box to the position you want on the report. It currently reads **text**. To change this, double-click to bring up the text requester. Delete the word **text** and type in the word or words of your choice. Each text box will accept up to 38 characters; letters, numbers, spaces, hyphens and underlines are all acceptable.

When the text has been typed in, select OK to place it into the report. The text box automatically becomes the right size for the text inside it. Text boxes can be moved around the screen in the same way as field boxes, by clicking and dragging. Text can be added into any section of the report.

Including Calculations

Calculations can be included in the report. Select **Field** from the top of the report definition screen. Next to the bottom of the list is the word **<Calculation>**. Select this and you get a box labelled **Calc1**. Place this box in the required position on the report layout. Double-click on the box to bring up the calculation requester.



From this, selecting **Calc** will produce the calculation definition requester with list of operators, field list and numeric keypad. Enter the required formula and select **OK**. Make sure that the field type for the calculation field is **Text** or **Numeric** depending on the result of the calculation, i.e. if it is a mathematical formula the result will be numeric, but if it is a string-handling operation, the result will be text. Selecting **OK** will incorporate the calculation into the report.

More than one calculation can be used in a report. Subsequent calculations will be labelled **Calc2**, **Calc3** etc.

Printing statistics in After sections

The statistical functions **Count**, **Sum**, **Min**, **Max**, **Mean** and **SDev** are all described in detail in **Appendix A**, but it is worth remembering that their arguments must always be enclosed in quotation marks; a few examples are mentioned here. In the "countries" example adding a calculation to the **After** group country section with the formula: **Count (" ")** would return 2 and 1. If it was added to the **After** report section it would return 3.

The Sum function is normally used for calculation totals, but it can also be used to calculate how many times a value of a field was inside an interval by using the conditional operator. Suppose you had a file containing information on people's ages. Putting the following calculations in the After report section would tell you how many were aged between 0 to 10, 10 to 20 and how many were older than 20:

```
Sum("age >= 0 && age < 10")  
Sum("age >= 10 && age < 20")  
Sum("age > 20")
```

Performance Considerations

Each statistical function you add means that an extra calculation must be performed for each record. If you only have a few hundred records this doesn't really matter, but if you have say 10 thousand records putting 20 statistical expressions in the report will require the evaluation of 200000 expressions. Note that if some of the argument strings to the statistical functions are the same, Twist will perform the evaluation only once, for example:

```
Sum("TotalAmount*0.15")  
Sum("TotalAmount")
```

Should be written as:

```
Sum("TotalAmount")*0.15  
Sum("TotalAmount")
```

The evaluation will be performed more quickly since the string argument is the same, hence it need only be calculated once and in the first example the multiplication by 0.15 is carried out for every record while in the second example it is only carried out once, when the result is printed in the After report section.

Another way of improving performance is to avoid unnecessary sorts. Suppose that in the example with countries, we only wanted to know how many cities were present in England and how many cities in Germany. The easiest way would be to group by country and use the Count() function. However it would be quicker not to use a group section but instead perform the following calculations in the report section:

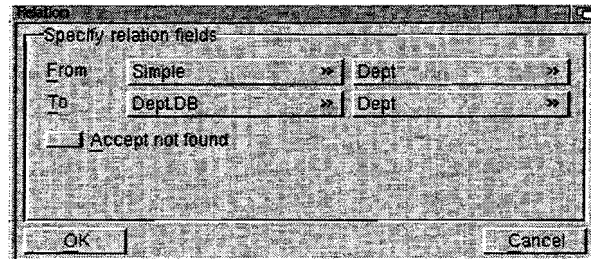
```
Sum(" Country == \"London\" ")  
Sum(" Country == \"Germany\" ")
```

Note the backslash being used when a quotation mark is present inside a string.

Linking two Files

It is possible to combine data from two different database files into one report. To do so there must be a related field in each file. That is a field which holds the same data, in the same format. In addition, that field *must* be indexed in the second file. That is, not the file in which the report is originally created, but the file which is to be brought into it.

To link in a second file, select **Field** from the top of the report definition screen. At the bottom of the field list is the legend <Change file>. Select this and then <New file> and you will get the relation requester:



On the top line it says **From** and the name of the current database file. Next to that is a box containing a question mark. Clicking on that box will produce a field list and you can select the field that you want to relate to the other file. The next line says **To** and has two boxes with question marks. Clicking on the first box will produce a list of all database files available. Select the one you want and click on the second question mark to produce the field list. If the related field in each file has the same name and structure, just selecting the second file name will place the field name into each row automatically.

If **Accept not found** is on and a record is found in the main file for which there is no corresponding record in the second file, the record will still be included in the report. Any fields for the second file will be blank. If **Accept not found** is off and a record is found in the main file for which there is no corresponding record in the second file, then the report will be aborted.

When the requester is complete, select **OK**. You will return to the report definition screen. Now when you select **Field** from the top, you will get a list of fields in the second file. You can select these and position them on the report layout, just as for fields from the main file.

To change from one file to the other for selecting fields, select <Change file> from the bottom of the field list and the file name from the subsequent list. This gives you the chance to change backwards and forwards from one file to the other as you refine your layout. It also gives you the option to relate to a new file.

Running the Report

Once the report layout is complete, save it by clicking in the top left-hand corner of the screen. This takes you back to the report screen. Before running the report, check that the output is directed to where you want it; Window, Printer, Clipboard or File. Then select **Run** and the report will be executed.

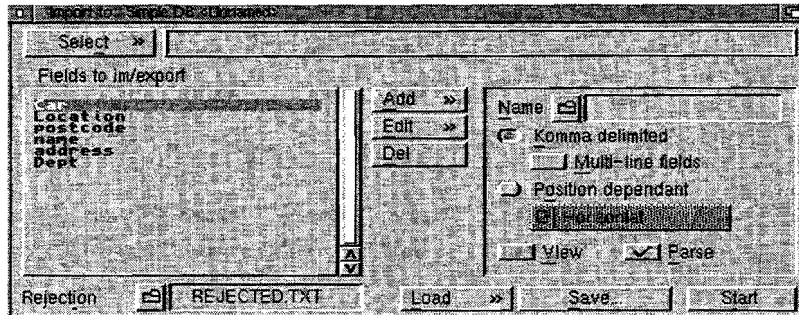
If it is directed to the screen, then you can view it first and still have the chance to print it later, when you are sure that it is laid out how you want and with the records you expected. When you run the report with output to window, it will be displayed on the screen and to clear the display, you click in the top left-hand corner of the report output screen. This takes you back to the report screen again. Running the report with output directed anywhere else, keeps you at the report screen. Once again, to leave this screen, click in the top left-hand corner of the window.

You return to the record screen. Now when you select **Report**, the name of the report you have set up will be listed, as well as the **Edit** option. To run the report as it is, select the report name. The report will run automatically, with output directed to whichever option was last selected. Selecting **Edit** allows you to set up a new report, or to edit existing ones.

The report can also be run directly from the Twist program screen, without needing to open the database file first. Either install an icon, in the same way that an icon was installed for the database program or by selecting **Project, Open** and then selecting the report from the file selector. The report file will have the same name as the database file, with the extension name1.R for the first report set up for that database, name2.R for the second report and so on.

Import/Export

The Twist Import and Export options allow new data to merged in or existing data to be exported to ASCII files. The Import/Export requester is used to control the export of data from Twist to a text file:



The **Select** field and button are used to define the records to process; this is used in the manner described in the report section. An existing query string may be used by selecting it from the **Select** pop-up, or a new one defined via the **Edit expression** selection on the pop-up.

The field list is shown in the left hand pane, Twist automatically adds all field names in the same order as they appear in the field definition. Alternatively fields may be added, edited or deleted via the **Add**, **Edit** and **Del** pop-ups - note that only the available fields are shown. In addition a **<Filler>** field may be added; this is used to add a null padding field, if required.

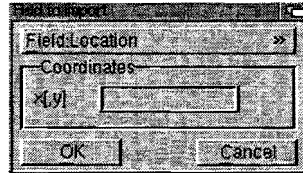
The data format is set in the right hand pane, either **Comma delimited** (often called a comma separated value, or CSV file), or **Position dependant** (where each field begins at a particular column or row position).

The comma separated file option uses one record per-line, with each field enclosed in quotes (""), separated by a comma (.). In addition if the **Multi-line fields** box is checked then any memo fields which have a newline character in them will be export across multiple lines.

The position dependent format uses fields either at a fixed horizontal, fixed vertical or fixed horizontal and vertical position (selected by the cycle gadget).

The **View** option allows the text file to be shown after export (or before import) in a Twist text editor window.

The position of each field may then be set by double-clicking on the required field in the field list, to open the position requester:



Export

During export the additional checkbox, **Delete exported records**, is available; this allows records in the database to be deleted after they have been successfully exported.

Import

During file import, the **Parse** check box is normally selected; when the import file is chosen Twist will look at the file chosen and attempt to deduce the file's format, saving you from the chore. After this the field list will show some additional characters:

- 9 The field is a text field, but numeric data was found in this field's position.
- a The field is a numeric field, but non-numeric data was found in this field's position.
- ? Twist could not decide on the format of the data.

The **Reject file** button allows a file to be set to which Twist will write records it finds whilst importing the data which do not match the field types.

MailMerge

Mailmerge is a term used to describe the operation of combining two files, to create one document, or a series of documents with the same layout. One of the files will be a text file and the other will be a database. The most common application of this procedure is to personalise standard letters, by merging individual details into the text of a letter, hence the term - mailmerge.

In order to create such a file, it is necessary to have two component files, the database and the text file. The text file can be created with any standard Amiga editor and contains certain control characters that Twist will recognise when performing the mailmerge.

Alternatively you can prepare a report file or an export file using Twist and then use the mailmerge facilities of your favourite word processor or desktop publishing program.

Preparing a Text File

As we have said, you can prepare the text for the mailmerge using any text editor - whenever you wish to include fields from a Twist database you must simply enclose the field references in ampersand (&) signs. If you wish to use an ampersand in your text you must use double-ampersand (&&).

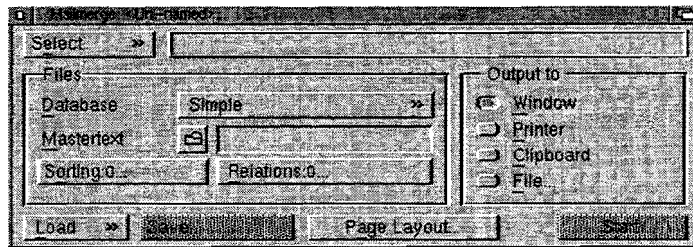
So your field references should look like `&filename.fieldname&`. If for example, you wanted to send a letter to each of your customers, addressing them by name and had those details in a database file, the first line of your letter might well look like this:

Dear `&customer.Title&` `&customer.Surname&`,

where `customer` is the name of the database and `Title` and `Surname` are the names of two of the fields in that database.

Creating a Mailmerge File

When Mailmerge is selected from the Report menu, the *Mailmerge* requester is displayed:



The **Select** line gives the expression which is used for selecting the records to be output; this works in the same way as all other Twist select buttons/boxes.

The **Database** pop-up sets the main database for the mail merge; if there are related files in the mail merge these may be set up via the **Relations** button, again this operates in the normal Twist way.

The **Mastertext** button sets the main file from which the text for the mail merge is read; this is the file which you have created using any standard text editor on the Amiga.

Running the Mailmerge

Once the database file has been chosen, the selection criteria entered, the related files (if any) set up and the master text file selected, the mailmerge is ready to go.

The output can be directed to either **Window**, **Printer**, **Clipboard** or **File**. Even if you want to print the documents, it is worth sending them to the screen first, in order to be able to check that the actual output corresponds to the expected output. Also the output can be edited if necessary in the same way that a report output can be edited from the screen.

Having selected the output destination of your choice, select **Start** to create the merged documents. A new document will be created for each record selected from the database.

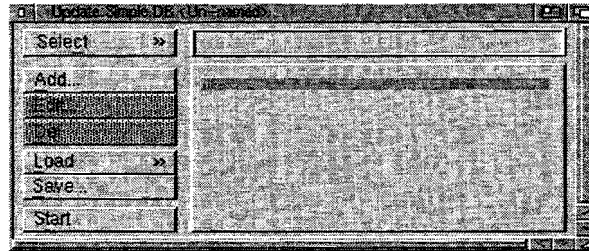
Saving the Mailmerge File

Once you are satisfied with the layout of the text and the selection of records, you may well want to save the mailmerge file, to be able to use it again in the future. To do this select **Save** from the mailmerge requester. You will be prompted for a name for the file. Type in the name and select **OK**. The file will be saved. To access the file again, to run it, or to edit it, select **Load** from the Mailmerge requester. Alternatively, it can be run from the Twist program screen.

To run a mailmerge operation directly from the Twist program screen, select **Project, Open**. From the file selector screen, select the mailmerge file. mailmerge files have the same name as the database file they are created with and the extension **.M** and a number depending on whether it is the first mailmerge file created with that database (name1.M) or the second (name2.M) and so on.

Update

An update is a sequence of calculations performed using new field values on a selected group of records. An update consists of a query string defining which records the calculation must be performed on and a list of field names and calculation formulae defining the new field values. If the update must access several files, a list of relations is also needed defining how the files are connected.



Some updates you will only want to perform once after you have changed the record structure while others might be useful to run on a regular basis, for instance to keep an inventory file up to date with recent sales. Such updates may be saved under a user-defined name and installed on the *Twist Icon* window for easy access.

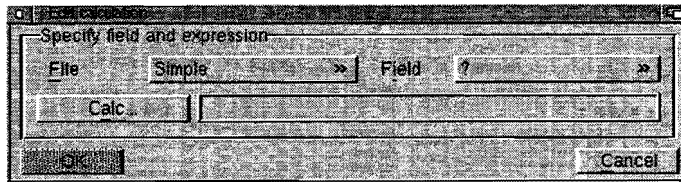
The update window Selection

An update can be performed on a selected group of records. Click on **Select** to open the *Select expression* box and define the query string. If you wish you may also enter the query string directly in the input box to the right of the **Select** button. If no query string is entered all records in the file will be included in the update.

Add/Edit/Delete

These are the standard buttons needed to edit the list of calculations or relations. W

Add and Edit open the *Edit calculation* requester :



Relations are defined in the same way as when used in reports; note that when the first calculation is added Twist will ask if you wish to use the same relations as defined in the form. Normally this will be the case:

The *Edit calculation* requester has the following buttons:

File

This defines in which file the field is located. By default this will be the current file. Choosing another file will usually require you to define a relation for that file as well.

Field

Select the field whose values you wish to change from a pop-up list. If the file contains more than 8 fields the pop-up list may be scrolled.

Calc

Click on Calc to open the calculation box or enter the calculation formula directly in the input box to the right of the button. This is the calculation which will be applied to every selected record in the database.

OK

Click OK when the field and the formula have been defined. If no calculation or field has been defined then the OK button is disabled.

Save

Saves the edited update specification. If the update specification is unnamed a unique name of up to 38 characters must be entered. The update file created will have the extension ?.U, where the ? is replaced by a number. Later you may wish to install the file on the *Twist Icon* window.

Load

Loads a previously saved update file. A pop-up list of updates appears to let you select the update to load. The last item in the list is <File Selector> which allows you to use the standard file selector to select the file.

Start

Starts the edited update. A update need not to be saved before it can be run. Depending on how many records need to be processed the update may take some time.

Changing values of fields that are indexed is always slower than using non-indexed fields because Twist must keep the index file(s) updated as well. However Twist may choose to rebuild the whole index file instead of updating each field value in the existing index. This is much quicker.

Warning. Be very careful what you do with updates. Twist can update hundreds of records in seconds. If the calculation formula was incorrect it may produce an unwanted result and you have no undo option. Therefore you should take a backup copy of your database file before you run any new update or at least duplicate a record and limit the update to that record (by entering the \$current selection criteria).

Appendix A

Functions & Expressions

In order to allow powerful manipulation of database fields, Twist includes a full expression evaluator and a set of types, together with a comprehensive set of functions, to allow complex computations to be performed.

Expressions

Twist's expression evaluator supports several types for its operands: integer, floating point and string. Integer supports only whole numbers, with a range of -2147483647 to 2147483647. Floating point supports fractional numbers to an accuracy of about 20 digits, up to numbers around 1e4900. Integer operands are automatically converted to floating point operands when required. For some operations a pseudo-boolean type is supported (true or false); strictly true has the numeric value 1 and false the numeric value 0, although any non-zero value is taken to be true.

The Twist expression evaluator uses the same symbols and precedence as that of the C language, from highest to lowest this priority is:

()	Parentheses
+ -	Unary plus and minus
!	Logical NOT
^	Power
* /	Multiplication, Division
+ -	Addition, Subtraction
#	The 'like' operator
> < >= <=	Greater than, Less than, Greater than or Equal, Less than or equal
= !=	Equal, Not Equal
&&	Logical AND
	Logical OR
?:	Conditional operator

Special types

Special characters in string constants

A string constant is always enclosed in quotation marks, such as "Hello". Sometimes it is necessary to include special characters inside a string, this is achieved by using the backslash symbol, \, followed by another character:

\ "	Defines a quotation mark inside a string so that "\"Hello\" he said" evaluates to "Hello" he said
\r	Carriage Return (CR, ASCII 13)
\n	Line feed or newline (LF, ASCII 10)
\f	Form feed (FF, ASCII 12)
\t	Tabulator (TAB, ASCII 9)
\\	Backslash (ASCII 92) "C:\\TWIST*.*" evaluates to C:\\TWIST*.*
\\x<hex>	Defines any character by its hexadecimal ASCII value: \\x1B Defines the character with the hexadecimal value \$41 which is the letter 'A'

Dates used in calculations

Twist stores dates as integers (namely the number of days A.D.). This means that you can use arithmetic on dates, thus:

Second_date - First_date

calculates the number of days which have elapsed between the two.

Function Parameters

Most functions need parameters. When Twist performs a syntax check the function list serves to validate that the correct number of arguments have been entered. All functions have a fixed number of operators. At run time (i.e. not when the formula is entered but when the function is actually used) the type of the parameter is also validated.

Examples of numeric parameters:

123	an integer
123.45	a floating point number
VAT	a field whose type is numerical
len(name)	the return value of another function
orderdate	a field whose type is Date

Examples of text parameters:

"Hello"	a string constant
name	a fieldname whose type is text
"He1"+"lo"	the text result of a calculation

Special functions exist to convert between text and numbers; Val(\$) and Str(9).

Arguments to statistical functions

Statistical functions can only be used in reports and even then, only in the sections After group and After report.

All statistical functions take a single string constant as their argument. This string must contain an expression which, when evaluated, returns a numerical value. The reason for this is quite complicated and is to do with the way in which Twist works internally. All you need to remember is to place quotations marks around the expression or field about which you wish to obtain statistical information.

One complication may arise when you wish to gather statistics using functions and operators that work on string constants. If the statistical expression string needs to have quotation marks *inside* the string, such quotation marks need to be preceded by the backslash character (so that Twist knows that this is not the end of the string).

The Operators

An operator is a mathematical or logical condition which determines the way in which an expression is processed.

(...)

Parentheses

Apart from being used to enclose parameters to functions, parentheses may also be used to modify the evaluation order of an expression, for example:

$$4*5+4 = 24$$

since the `*` operator has a higher priority than the `+` operator; if this is not desired, then parentheses must be used:

$$4*(5+4) = 36$$

Unary + and -

Sign

The unary `+` and `-` operators change the sign of the associated argument; note that unary `+` is included only for completeness since `4` and `+4` have the same value.

!

Logical not

The logical not operator inverts the truth state of its operand, thus an expression that was true becomes false and vice versa. Any value other than 0 is true, thus `!5` evaluates to 0 and `!0` evaluates to 1.

^

Power

The power operator causes the left hand operand to be raised to the power of the right hand operand, for example

$$2^8 = 256$$

*** and /**

Multiplication, Division

The `*` operator performs normal multiplication, for example:

$$5*8 = 40$$

The `/` operator performs normal division:

$$40/5 = 8$$

Note that both operators use the type of the operands, this means that for example, `5/2` will yield 2 (since both 5 and 2 are integers), whereas `5.0/2` will give 2.5 (since 5.0 is floating point).

+ and -

Addition, Subtraction

The + operator, with numeric (floating point or integer) arguments performs normal addition of the two numbers. Similarly the - operator performs normal subtraction of two numbers.

Also the + operator may be used with two string arguments in order to concatenate them end-to-end, for example

```
"Hello" + " " + "World" = "Hello World"
```

#

Like

The like operator can only be used on text operands, where it performs a case-insensitive (upper and lower case are treated as the same) comparison.

Optionally, the second operator may contain the wild card symbol * at the beginning and/or at the end of the string which matches any number of characters, for example:

```
"Hello world" # "hello world"
```

```
"HELLO WORLD" # "*lo world"
```

```
"Hello" # "*e*"
```

Are all 'true' expressions. Of course, directly comparing string constants as shown above is not very useful - usually the first operand will be a text field in your database or the result of a calculation which has been converted into text.

>

Greater than

<

Less than

> =

Greater than or equal

< =

Less than or equal

The relational operators (as in numeric relations, not database relations), evaluate to true if the left hand operand relates to the right hand operand in the manner specified, for example:

```
5 > 2
```

evaluates to 'true' since 5 is greater than 2.

In addition strings may be used with these operators, yielding a result based on the relative ordering of the letters; note that this is case-insensitive, hence:

"a" < "B"

evaluates as 'true'.

= =
!=

Equal to
Not equal to

The equality operators compare two operands (which can be either numeric or text) and return either true or false. For strings, the comparison *is* case significant, cf. the # operator which performs a case-insensitive string comparison.

&&

Logical and

Returns true if both operands are true; remember that any numerical value is true if it is non-zero, for example:

5 && 1 is true
0 && 5 is false
(0==0) && (!0) is true

||

Logical or

Returns true if one or both of the operands is true, for example

5 || 1 is true
0 || 1 is true
0 || 0 is false

?:

Conditional

The conditional operator is used to evaluate one operand and then selectively evaluate one of the two remaining operands; note that in total *three* operands are required:

condition ? expr1 : expr2

The condition is evaluated first, if true (non-zero) the value of expr1 is returned otherwise the value of expr2 is returned, for example:

4 ? 10 : 100 = 100
0 ? " Yes" : "No" = "No"

A more complex example:

`VAT>0 ? "Taxed" : "Not Taxed"`

returns either "Taxed" or "Not Taxed" depending on whether the expression `VAT>0` is true or false.

As with all other expressions you can put several together to form more complex expressions such as:

`CustGroup==1 ? "VIP":(CustGroup==2 ? "Important":"Ordinary")`

which will return either "VIP", "Important" or "Ordinary".

The Functions

The functions supported within Twist are where much of its real power lies. Each of the functions takes a number of parameters, the type of which may be either numeric (integer or floating point) or string; within this documentation and the Twist function list within the program, a numeric parameter is indicated as 9, a string parameter as \$ and a field name as *.

In the following reference Twist function names are shown in mixed case (ACos for example), the actual case of these is unimportant to Twist - acos, ACOS and ACos are all equally valid.

Abs(9)

Absolute value

The Abs function computes the absolute value of its numeric argument, for example:

`abs(4) = 4`

`abs(-4) = 4`

ACos(9)

Arc-Cosine

The ACos function computes the normal arc-cosine function (inverse cosine) of its argument (in the range -1...1), returning a value in radians, for example:

`ACos(0) = 1.570796`

i.e. $\pi/2$.

AnyLikelyness(\$)

AnyLikelyness match

The AnyLikelyness function matches its argument against all text fields in the current record returning the highest score of all fields. This function is used implicitly by the free text search command where matches are those which return a positive value.

ASin(9)

Arc-Sine

The ASin function computes the normal arc-sine function (inverse sine) of its argument (in the range -1...1), returning a value in radians, for example:

ASin(-1) = -1.570796

i.e. $-\pi/2$.

ATan(9)

Arc-Tangent

The ATan function computes the normal arc-tangent function (inverse tangent) of its argument (in the range $-\infty \dots \infty$), returning a value in radians, for example:

ATan(1) = 0.785398

i.e. $\pi/4$.

Cos(9)

Cosine

The Cos function computes the normal cosine function of its argument, in radians, for example:

Cos(1.570796) = 0

Count(\$)

Count of values in group

Count is a statistical function which can be used in reports to return the number of occurrences of a given expression so far; the argument passed to Count *must* be a string.

The way in which Count works may not be immediately obvious and depends on the context of the string expression. At the beginning of a report all count expressions are reset to zero, the count then increases each time the given expression returns a 'defined' numerical value and is performed once for each record in the report.

One use for Count is to return the number of records in a database. To do this you would use Count ("0"). This works because the expression is evaluated once for each record and "0" will always return a defined value (namely zero, which in this context is *not* the same as false).

DateD(9)

Extract day of date

The DateD function extracts the day from a numeric value used to represent a date, for example:

DateD(727928) = 31

DateM(9)

Extract month of date

The DateM function extracts the month from a numeric value used to represent a date, for example:

DateD(727928) = 12

DateStr(9, \$)

Convert date value to string

The DateStr function converts its numeric argument to a string formatted in a manner specified by the based string parameter, if this string is the empty string, "", then the default format is used. The format string has the same form as that used in other string formatting areas in the order DMYS, MDYS or YMDS, where D, M, Y and S are:

D = "d" (always)

M = "m2", "m3" or "m" ("06", "Jun", "June" respectively)

Y = "y2" or "y4" ("94" or "1994" respectively)

S = "/", "-" or "." ("/", "-" or "." separators)

For example,:

dm2y2/ 2/06/94

y4md. 1994 June 2

DateVal(\$)

Convert string to date value

DateVal converts a date string into a numerical value; the sequence of days, month and year should be the same as in the default date format. If however the month is given as a text value (e.g. Jan 10 1993) Twist will attempt to extract it anyway, for example:

DateVal("31.12.92") = 727928

DateY(9)**Extract year of date**

The `DateY` function extracts the year from a numeric value used to represent a date, for example:

`DateY(727928) = 1992`

DateYDiff(9, 9)**Compute difference in years**

The `DateYDiff` function subtracts its second argument from the first and returns the difference as a number of years; typically this is used as:

`DateYDiff(DateOfBirth, Today())`

to calculate the age of someone given a date of birth.

Deg(9)**Convert radians to degrees**

The `Deg` function converts its argument, in radians, into degrees; it is equivalent to $a * 180/\pi$, for example:

`Deg(1.570796) = 90`

FillStr(\$, 9)**Create string of characters**

The `FillStr` function creates a string of a length defined by the second argument, the content of which is determined by the first argument, which will usually be a string one character in length, for example:

`FillStr("-", 9) = "-----"`

`FillStr("#*", 9) = "*#*#*#*#*#*"`

FindText(\$)**Find string in record text fields**

`FindText` searches all text fields in the current record for the occurrence of the given text fragment, returning the number of the first field in which the fragment was found; note that the first field in a record is field 0. If the function fails to find an occurrence of the fragment, then -1 is returned.

This function is used implicitly by `Twist` for the free text search option in the query window.

Fix(9)**Extract integer part of number**

The `Fix` function returns the integer part of a floating point number, for example:

`Fix(43.323) = 43`

FixLStr(\$, 9)

Extract fixed length string

The **FixLStr** function extracts the left- or right-most number of characters specified by its numeric argument (left-most if positive, right-most if negative) from its string argument. If the string is shorter than the requested length the returned string will be padded to the correct length with spaces.

Frac(9)

Extract fractional part of number

The **Frac** function returns the fractional part of a floating point number, for example:

Fix(43.323) = 0.323

InStr(\$, \$)

Find sub-string within string

InStr searches the first string for the first occurrence of the second, returning a number which represents the character position inside the first string where the match was found (the first character is considered to be character 0). If the function fails to find a match, then -1 is returned; note that the search is performed without respect to the case of letters. For example:

InStr("Hello World", "ll") = 2

InStr("Hello World", "HELLO") = 0

InStr("Hello World", "Good bye") = -1

IsUndef(9)

Determine if argument is undefined

IsUndef determines if its argument is an undefined expression; an undefined field is one in which no value has been entered by the user, an undefined expression is one which consists entirely of undefined fields. Note that a text field can never be undefined.

LastRVal(*)

Determine previous record content

Given a field name, **LastRVal** determines the field content for the *previous* record; this is particularly useful in After group report sections to find the field value of the last record or to compare the current value with the previous value.

The function may only be used in reports. For the first record in the report the previous record is an empty record.

Left(\$, 9)**Extract left-most characters**

The **Left** function extracts the number of left-most characters specified by the numeric argument from the string. If the source string is shorter than the numeric argument then the resulting string will not be as long as expected.

For example:

Left("Hello World", 5) = "Hello"

Len(\$)**Find length of string**

Returns the number of characters in the given string, for example:

Len("Hello World") = 11

Len("Hel"+"lo") = 5

Likelyness(\$, \$)**Find likelyness of match**

The **Likelyness** function computes a score for how likely it is that the second argument matches the first. The first argument will normally be a text field which is to be checked, the second argument is the comparison string, which *must* be in upper case.

Max(\$)**Obtain maximum value from group**

Max is a statistical function used in reports to find the maximum value of the quoted expression during the generation of a group.

For example, to find the lexically longest string in a field, the expression:

Max("Len(FieldName)")

could be used.

Mean(\$)**Obtain arithmetic mean of group**

This statistical function returns the mean (average) value of all records given by the quoted expression in the group whilst generating a report.

MidStr(\$, 9, 9)**Extract portion of string**

MidStr extracts a sub-string from its string argument, starting at the character position specified by its first numeric argument, over a length given by the second numeric argument; if either argument is out-of-range the they are adjusted as required. For example:

```
MidStr("Hello World", 0, 5) = "Hello"
```

```
MidStr("Hello World", 9, 100) = "ld"
```

```
MidStr("Hello World", 100, 5) = ""
```

Min(\$)**Obtain minimum value from group**

Min is a statistical function used in reports to find the minimum value of the quoted expression during the generation of a group.

For example, to find the lexically shortest string in a field, the expression:

```
Min("Len(FieldName)")
```

could be used.

Mod(9, 9)**Modulus**

Mod finds the modulus of two numbers (i.e. the remainder after integer division); note that both arguments are implicitly converted to integers prior to calculation, hence any fractional value will be lost. For example:

```
Mod(7, 4) = 3
```

```
Mod(7.13, 4.9) = 3
```

Randomize(9)**Obtain random number**

The Randomize function returns a random number between 0 and its argument (inclusive), for example:

```
Randomize(10)
```

returns values between 0 and 10.

RecordID(\$)**Obtain record number**

Returns the current record's internal identification number for a given filename; if the string argument is the empty string (" ") then the current file is used.

The record ID is an integer that is assigned to a record when it is created. All records in a file have different identification numbers and even if the record is changed or the whole file is re-organised, the ID for each record remains unchanged. Record IDs are often used in field calculations to give a record a unique number, this is especially useful when working with relations between records.

RecordNumber(\$)

Obtain record position

Returns the physical position of a record within a file; when records are created, the space reserved for a previously deleted record will be used first, thus it is not always true that new records will have higher record numbers. If a file is re-organised, all deleted records are overwritten and the records are renumbered.

Right(\$, 9)

Extract right-most characters

The Right function extracts the number of right-most characters specified by the numeric argument from the string. If the source string is shorter than the numeric argument then the resulting string will not be as long as expected.

For example:

Right("Hello World", 5) = "World"

SDev(\$)

Standard deviation

SDev is a statistical function used in reports to find the standard deviation of the quoted expression during the generation of a group; the standard deviation is a numerical value indicating the spread of the sampled fields - if all encountered values were equal, then SDev would return 0.

Sin(9)

Sine

The Sin function computes the normal sine function of its argument, in radians, for example:

Sin(1.570796) = 1

Sqr(9)

Square-root

Returns the square root of its numerical argument; note that if the argument is negative, 0 is returned, for example:

Sqr(25) = 5

Sqr(10.89) = 3.3

Str(9, \$)

Convert to string

Converts a number into the corresponding string; the string argument gives the required format, or "" for the default format. The format string has the form <Sign>[<Filler>]<Digits>['.'<Decimals>] (note that [] are used to indicate optional parameters), the parameters are:

<Sign>:

' '	Positive numbers are space prefixed, negative numbers '-' prefixed
'+'	Positive numbers are '+' prefixed, negative numbers '-' prefixed
'-'	Positive numbers have no prefix, negative numbers '-' prefixed

<Filler>:

	(Nothing). The value is right-aligned by padding with spaces
'*'	The value is right-aligned and padded with asterisks
'<'	The value is left-aligned with no additional padding

<Digits> is the number of digits the number may contain; this is used for calculating the field width. Values from 1...18 should be used.

<Decimals> is the number of places of decimals to generate, preceded by a period, ., legal values are 1...8.

Sum(\$)

Sum is a statistical function used in reports to find the sum of the quoted expression during the generation of a group.

Tan(9)

Tangent

The Tan function computes the normal tangent function of its argument, in radians, for example:

$\text{Tan}(0.785398) = 1$

Time()

Return system time

The Time function returns the system time expressed in seconds past midnight.

TimeStr(9) Convert number of seconds to string

The TimeStr function returns a string in the format HH:MM:SS which corresponds to its numeric argument, a number of seconds.

TimeVal(\$) **Convert a string to a number of seconds**

The TimeVal function converts a string of the form HH:MM:SS into a corresponding number of seconds; in addition a number of shorthands for the time are supported and automatically converted to a usable format.

Today() **Obtain current day 'value'**

Returns the current day expressed as a numeric value.

ToLower(\$) **Return lowercased string**

The ToLower function converts its argument to the lower case equivalent, for example:

ToLower("Hello World") = "hello world"

ToUpper(\$) **Return uppercased string**

The ToUpper function converts its argument to the upper case equivalent, for example:

ToUpper("Hello World") = "HELLO WORLD"

Val(\$) **Convert string to number**

Val converts a text string containing digits into a numeric value, for example:

Val("100.1") = 100.1

WeekDayStr(9) **Return day-of-week string**

The WeekDayStr function returns a string representing the day of the week given by its numeric argument (which should be a 'date' value), for example:

WeekDayStr(727928) = "Thursday"

Appendix B

ARexx

Twist's ARexx Interface

This information is brief; please refer to the README file for more information on using ARexx with Twist.

Naming Ports

"TWIST" is the main port name. This is the default port used for simple ARexx scripts. With the default port it is possible to work with multiple files, but to use nested SELECTALLs or accessing Twist with concurrent scripts requires forced port names.

To force another port name use "OPEN <filename> PORT FORCE" which returns the new port. This port is the handle to a "project". A project contains a list of open files, a current file, a data structure associated with a SELECTALL command and a window pointer.

All ports different from the default port "TWIST" will be named TWIST.<X>. Where X is a number starting from 1. The script can of course name the project itself.

Using multiple projects allows scripts to perform nested SELECTALLs thus allowing multiple scripts to run simultaneously. See SELECTALL and OPEN for further information.

Quotation Marks

There are 3 different levels of quotations marks, those used by ARexx in general , Twists ARexx command interpreter and Twists Query interpreter

Quotation Marks used by ARexx in general

Suppose Twist had a command named TestQuote which required no arguments. In a script the command could be called by:

ADRESS TWIST

'TestQuote' or

ADRESS TWIST

"TestQuote" or

ADRESS TWIST

TestQuote

In every case, Twist would receive the string TestQuote, i.e. the quotation marks had been stripped by ARexx.

The Command Interpreter

Twists ARexx command interpreter allows filenames to contain spaces (although this is normally not recommended and in many cases not even allowed). To open such a file we can use the OPEN command, however given that ARexx strips quotation marks care must be taken to avoid the wrong filename being passed. If for example we wanted to open the file My mailmerge file Twist would need the command,

OPEN "My mailmerge File"

The quotation marks would first be stripped by ARexx then Twist would interpret the filename. If the quotation marks were not present, ARexx would only pass My to Twist. Thus the quotation marks are needed to keep the filename from being interpreted as 3 separate arguments. Had the file been named My_Mailmerge_File or mymailmergefile then quotation marks wouldn't be necessary.

Twist's Query interpreter

The query interpreter is used mainly for the SELECTALL command to specify which records to select from the database. A typical query string could be "UnitCost == 1000". In the SELECTALL command the query string must be given as a single argument, thus you would need to pass the string as,

```
SELECTALL WHERE "UnitCost == 1000"
```

again the quotation marks are needed because of ARexx.

Appendix C

Technical Support

Twist comes with 30 days free technical support, starting from the date of registration; therefore you should send in your registration card quickly. Technical support is available by telephone during our Technical Support Hour, by letter or by fax.

Should you wish to receive extended technical support, please complete the relevant sections on the registration card, indicating whether you would like to take up the Silver or the Gold service.

In addition to your name, address and postcode (very important for UK customers), we need payment details before we can accept your extended registration. You can pay by credit card (Mastercard, Eurocard, Access, Visa etc.), UK debit card (Switch, Connect etc.), Eurocheque, UK cheque or Postal Order.

You may have already registered another HiSoft product under our Gold or Silver service; in this case, there is no need to fill out the payment section.



*Twist 2 is created by
Mermaid Group, Denmark
and published in the English
language by HiSoft, UK.*

HiSoft
High Quality Software